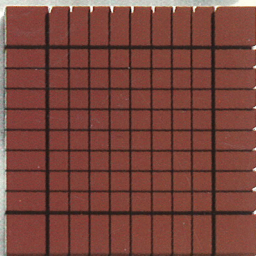
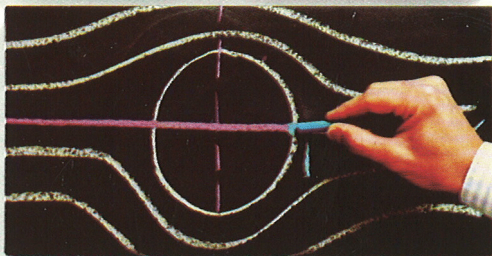
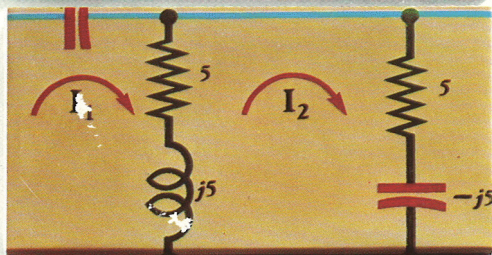
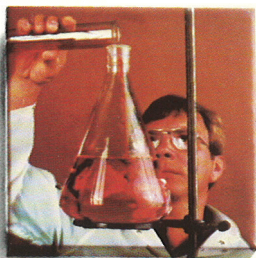
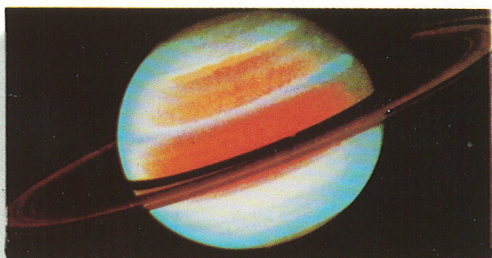


HEWLETT-PACKARD

HP-15C

BRUGERHÅNDBOG OG
EKSEMPELSAMLING





HP-15C

Brugerhåndbog og programmeringsvejledning

00015-90010

Introduktion

Tillykke! Uanset om det er første gang De skal arbejde med en lomme-regner fra Hewlett-Packard eller De er en erfaren HP-bruger, vil De opdage, at HP-15C ikke finder sin lige blandt lommeregnerne. Ud over kontinuerlig hukommelse og et lavt strømforbrug giver HP-15C's super-avancerede teknologi Dem følgende fordele:

- 448 bytes programhukommelse (én eller to bytes pr. instruktion) og avancerede programmeringsmuligheder – herunder betingede og ubetingede forgreninger, underprogrammer, flag og fejlretning.
- Fire avancerede matematiske rutiner: beregninger med komplekse tal, matrixberegninger, løsninger af ligninger og numerisk integration.
- Direkte og indirekte lagring i op til 67 dataregistre.
- Batterier med lang levetid.

Denne bog er skrevet så De skulle kunne få glæde af den uanset hvor meget eller hvor lidt De kender til lommeregnerne. Første kapitel, "HP-15C som regnemaskine", dækker alle basisfunktionerne. Hvert af afsnittene i andet kapitel, "Programmering", er opdelt i tre underafsnit: "Mekanikken", "Eksempler" og "Yderligere information", således at det bliver lettest muligt for brugere med forskellig baggrund at finde den information, de har brug for. I det sidste kapitel, "Avancerede funktioner", bliver de 4 avancerede matematiske rutiner beskrevet i nærmere detaljer.*

Før De starter på disse afsnit, vil det nok være bedst, hvis De tager Dem tid til at gennemgå "HP-15C: En problemløser", der starter på side 12 og giver Dem lidt praktisk erfaring i brug og programmering af HP-15C.

* Det er helt klart ikke nødvendigt for Dem at gennemgå kapitlerne I og II, hvis De som erfaren bruger ønsker at bruge de avancerede matematiske rutiner. Dog kræver brugen af **SOLVE** og  kendskab til programmering af HP-15C.

De forskellige tillæg dækker flere af detaljerne omkring brugen af lommeregneren, og der bliver desuden gjort rede for servicebetingelser og lignende praktiske forhold.

Lige inden stikordsregistret finder De en udførlig oversigt over alle funktionerne. Denne kan dels bruges til hurtig reference, dels til at finde det eller de afsnit i brugerhåndbogen, hvor funktionen omtales.

Hewlett-Packard har udgivet *HP-15C Advanced Functions Handbook*, som indeholder eksempler på anvendelse af de avancerede matematiske rutiner samt tekniske beskrivelser af rutinernes virkemåde.

Indholdsfortegnelse

HP-15C: En problemløser	12
Et hurtigt kig på ENTER	12
Manuel problemløsning	13
Programmeret problemløsning	14
Kapitel I: HP-15C som regnemaskine	17
Afsnit 1: Grundbegreberne	18
Tænd/Sluk – Afbryder	18
Tastfunktioner	18
Primære og alternative funktioner	18
Forvalgstaster	19
Fortegnsskift	19
Indtastning af eksponenter	19
"CLEAR"-tasterne	20
Nulstilling af lyspanel: CLx og 	21
Beregninger	22
Funktioner af én variabel	22
Funktioner af to variable og ENTER	22
Afsnit 2: Numeriske funktioner	24
Pi	24
Funktioner til ændring af det udlæste tal	24
Funktioner af én variabel	25
Generelle funktioner	25
Trigonometriske funktioner	25
Omregning af tid og vinkler	26
Omregning mellem grader og radian	27
Logaritmefunktioner	27
Hyperbolske funktioner	28
Funktioner af to variable	28
Potensopløftning	29
Procentregning	29
Omregning mellem polære og rektangulære koordinater	30
Afsnit 3: Den automatiske rullestak, SIDSTE X og dataregistre ...	32
Den automatiske rullestak og stakmanipulation	32
Funktioner til manipulering af stakken	33

SIDSTE X-registret og LSTx	35
Funktioner og rullestakken	36
Indtastningsrækkefølge og ENTER	37
Beregning af udtryk med parenteser	38
Konstantaritmetik	39
Brug af dataregistre	42
Lagring og fremkald af tal	42
Nulstilling af dataregistre	43
Direkte og omvendt registeraritmetik	43
Overløb og underløb	45
Opgaver	45
 Afsnit 4: Statistiske funktioner	 47
Sandsynlighedsregning	47
Pseudotilfældige tal	48
Summation af statistikdata	49
Korrektion af summerede data og datapar	51
Middelværdi	52
Standardafvigelse	53
Lineær regression	54
Lineære estimater og korrelationskoefficienter	55
Andre anvendelser	56
 Afsnit 5: Lyspanelet og den kontinuerlige hukommelse	 57
Udlæsningsformater	57
Fastkomma	57
Flydende tals repræsentation	58
Metrisk notation	58
Udlæsning af hele mantissen	59
Afrundingsfejl	59
Lyspanelet som indikatorpanel	59
Symbolrækken	59
3-cifferinddeling	60
Fejlmeddelelser	60
Overløb og underløb	60
Batteriindikator	61
Kontinuerlig hukommelse	61
Status	61
Nulstilling af den kontinuerlige hukommelse	62

Kapitel II: Programmering	63
Afsnit 6: Grundlæggende programmering	64
Mekanikken	64
Sådan skrives et program	64
Sådan lagres et program	64
Stop i selve programmet	66
Kørsel af et program	66
Indlæsning af data	67
Programhukommelsen	68
Eksempel	69
Yderligere information	72
Programinstruktioner	72
Tastkoder	73
Splitning af lager	73
Programgrænser	76
Uventede stop	77
Forkortet indtastning	77
USER-status	77
Polynomier og Horner's metode	78
Funktioner, som ikke kan programmeres	79
Opgaver	79
Afsnit 7: Editering	81
Mekanikken	81
Hop til bestemt linie i programhukommelsen	81
Sletning af programlinier	82
Indsættelse af programlinier	82
Eksempler	82
Yderligere information	84
Linievis udførelse	84
Programpilens placering	85
Indsættelse og sletning af programlinier	86
Initialisering som led i programkørsel	86
Opgaver	86
Afsnit 8: Forgreninger, løkker og flag	89
Mekanikken	89
Forgrening	89
Sammenligningsoperationer	90

Flag	91
Eksempler	92
Forgrening og løkkestruktur	92
Flag	94
Yderligere information	96
Go To - ubetinget hop	96
Løkker	97
Betinget hop	97
Flag	98
Systemflagene: Flag 8 og 9	99
 Afsnit 9: Underprogrammer	100
Mekanikken	100
Kald af underprogram og returhop	100
Kald af underprogram fra underprogram	101
Eksempler	101
Yderligere information	104
Returhoppet	104
Underprogrammer i flere niveauer	104
 Afsnit 10: Indeksregistret og løkker	105
I og (I)	105
Direkte kontra indirekte datalagring med indeksregistret	105
Indirekte programkontrol ved hjælp af indeksregistret	106
Løkker	106
Mekanikken	106
Lagring og fremkald med indeksregistret	106
Aritmetik med indeksregistret	107
Ombytning med X-registret	107
Indirekte forgrening med I	107
Indirekte kontrol af flag med I	108
Indirekte kontrol af udlæsningsformatet med I	108
Løkker med tællere: ISG og DSE	108
Eksempler	110
Registeroperationer	110
Løkkekontrol med DSE	111
Kontrol af udlæsningsformatet	113
Yderligere information	114
Indhold af indeksregistret	114

ISG og DSE	115
Indirekte kontrol af udlæsningsformatet	115
Kapitel III: Avancerede funktioner	117
Afsnit 11: Beregninger med komplekse tal	118
Den komplekse stak og kompleksstatus	118
Oprettelse af den komplekse stak	118
Deaktivering af kompleksstatus	119
Komplekse tal og stakken	119
Indtastning af komplekse tal	119
Stakløften i kompleksstatus	122
Manipulering af den reelle og den komplekse stak	122
Fortegnsskift	122
Sletning af et komplekst tal	123
Indtastning af et reelt tal	126
Indtastning af et imaginært tal	127
Lagring og fremkald af komplekse tal	128
Operationer med komplekse tal	128
Funktioner af én variabel	129
Funktioner af to variable	129
Funktioner til manipulering af stakken	129
Sammenligningsoperationer	130
Komplekse resultater af operationer på reelle tal	130
Omregning mellem polære og rektangulære koordinater	131
Opgaver	133
Yderligere information	134
Afsnit 12: Matrixregning	136
Matrixdimension	138
Dimensionering af matricer	139
Udlæsning af matrixdimension	140
Ændring af matrixdimensionen	140
Lagring og fremkald af matrixelementer	141
Lagring og fremkald af alle elementer i rækkefølge	141
Kontrol og ændring af enkelte matrixelementer	143
Lagring af samme tal i alle matricens elementer	145
Matrixoperationer	145
Matrixdeskriptorer	145
Resultatmatricen	146

Kopiering af en matrix	147
Operationer på én matrix	147
Skalare operationer	149
Aritmetiske operationer	151
Matrixmultiplikation	152
Løsning af ligningen $\mathbf{AX} = \mathbf{B}$	154
Beregning af restmatrix	157
Brug af LU -faktoriserede matricer	158
Beregninger med komplekse matricer	158
Lagring af elementerne i den komplekse matrix	159
De komplekse transformationer	162
Invertering af en kompleks matrix	163
Multiplikation af komplekse matricer	164
Løsning af den komplekse ligning $\mathbf{AX} = \mathbf{B}$	166
Forskellige operationer med matricer	170
Dataregisteroperationer med matrixelementer	170
Matrixdeskriptorer i indeksregistret	171
Sammenligningsoperationer med matrixdeskriptorer	171
Stakoperationer ved matrixoperationer	172
Brug af matrixoperationer i programmer	173
Oversigt over matrixfunktioner	174
Yderligere information	177
 Afsnit 13: Løsning af ligninger	178
Brug af SOLVE	178
Forgæves søgning efter nulpunkter	184
Startværdier	186
Brug af SOLVE i programmer	189
Begrænsninger i brugen af SOLVE	190
Krav til lagerplads	190
Reference til yderligere information	190
 Afsnit 14: Numerisk integration	191
Brug af ∫	191
Nøjagtigheden af ∫	197
Brug af ∫ i et program	200
Krav til lagerplads	200
Reference til yderligere information	201
 Tillæg A: Fejlmeddelelser	202

Tillæg B: Stakløft og SIDSTE X	206
Afslutning af talindtastning	206
Stakløft	206
Operationer, der blokerer stakløften	206
Operationer, der aktiverer stakløften	207
Neutrale operationer	208
SIDSTE X	209
Tillæg C: Splitning	210
Hukommelsen	210
Registre	210
Udlæsning af aktuel splitning med MEM	212
Splitning	212
Brugen af DIM (i)	212
Grænser for splitningen	213
Programhukommelse	214
Automatisk splitning	214
Programinstruktioner på 2 bytes	215
De avancerede rutiners pladskrav	215
Tillæg D: Detaljeret gennemgang af SOLVE	217
Mekanikken bag SOLVE	217
Nøjagtighed	219
Fortolkning af resultater	223
Ligninger med flere løsninger	230
Begrænsning af beregningstiden	235
Begrænsning af antal iterationer	235
Begrænsning af nøjagtighed	236
Reference til yderligere information	236
Tillæg E: Detaljeret gennemgang af ↵	237
Mekanikken bag ↵	237
Nøjagtighed, usikkerhed og beregningstid	238
Usikkerhed og udlæsningsformat	242
Forhold der kan føre til forkerte resultater	245
Forhold der forlænger beregningstiden	250
Udlæsning af foreløbig tilnærmelse	252
Reference til yderligere information	254
Tillæg F: Batteri, service og udbedring af mangler	255
Batterier	255

Batteriindikator	256
Udskiftning af batterier	256
Selvtest	259
Udbedring af mangler	260
Hvad der dækkes af garantien	260
Hvad der ikke dækkes af garantien	261
Specifikationer	261
International serviceorganisation	261
Oplysning om garantiforhold	264
Forsendelsesvejledning	264
Yderligere information	264
Temperaturspecifikationer	264
 Oversigt over funktionstaster	 265
Dataregisterfunktioner	265
Forvalgstaster	266
Hyperbolske funktioner	266
Indeksregister, kontrol af	266
Komplekse funktioner	266
Logaritme- og eksponentialfunktioner	267
Matematiske funktioner	267
Matrixfunktioner	267
Omregningsfunktioner	269
Procentfunktioner	269
Sandsynlighedsregning	269
Stakmanipulation	270
Statistik	270
Talindtastning	271
Trigonometriske funktioner	271
Udlæsningsformatet, kontrol af	271
Ændring af det udlæste tal, funktioner til	271
 Oversigt over programmeringstaster	 273
 Stikordsregister	 276
 HP-15C's tastatur og den kontinuerlige hukommelse	 Omslagets side 3

HP-15C: En problemløser

HP-15C programmerbar, avanceret lommeregner er en effektiv problemløser, der kan tages med næsten overalt. Dens kontinuerlige hukommelse bevarer data og programmer, indtil de slettes af Dem. Selv om HP-15C har så avancerede funktioner, kan den programmeres uden forudgående kendskab til programmering og programmeringssprog.

Det lave strømforbrug er en væsentlig fordel ved HP-15C. Det betyder, at De ikke skal tænke på oplader, lysnetkontakter, ledninger og bestandige batteriskift. Strømforbruget er så lavt, at De kan regne med mindst 6 til 12 måneders normal brug på ét sæt batterier. Og HP-15C har en batteriindikator, der giver signal i god tid inden det er nødvendigt at skifte batterier.

HP-15C sparer desuden energi med den automatiske batterisparer, der slukker for lyspanelet, når maskinen ikke har været brugt i nogle minutter. Men De behøver ikke at bekymre Dem om programmer og data; den kontinuerlige hukommelse bevarer alt præcis som før der blev slukket for lyspanelet.

Et hurtigt kig på **ENTER**

Deres lommeregner fra Hewlett-Packard arbejder med et indtastningssystem, der adskiller sig en del fra det, som bliver brugt på de fleste andre lommeregner. Forskellen koncentrerer sig om **ENTER**, og systemet har gennem alle de år, hvor Hewlett-Packard har fabrikeret lommeregner, vist sig at være det mest effektive og hurtigste af alle kendte indtastningssystemer. Men lad os se på et par eksempler.

Lad os f.eks. se på de aritmetiske funktioner. Først skal vi have et tal ind i maskinen. Er der tændt for den? Hvis ikke, så tryk på **ON**. Er lyspanelet nulstillet? Hvis ikke, så tryk **9** **CLx**, dvs. tryk på **9** og derefter på **←**.^{*} Når De regner aritmetisk, skal De først indtaste det første tal

^{*} De vil bemærke, at de fleste af tasterne har tilknyttet 3 funktioner. Funktionen på toppen af tasten aktiveres efter tryk på selve tasten; for at aktivere funktioner, der er trykt med gult eller blå, skal De først bruge den gule eller blå forvalgstast, og dernæst trykke på funktionstasten.

og trykke på **ENTER** for at afslutte indtastningen, dernæst indtaste det andet tal og trykke på tasten for den ønskede operation. Herved afsluttes indtastningen af det andet tal, og resultatet af operationen mellem de to tal bliver med det samme vist i lyspanelet.

I denne brugerhåndbog har vi brugt udlæsningsformatet **FIX** 4 (resultater afrundes til 4 decimaler), medmindre der direkte gøres opmærksom på andet. Hvis Deres lommeregner ikke viser 4 decimaler, skal De blot trykke **f** **FIX** 4.

Manuel problemløsning

Gennemregn følgende enkle opgaver. Det er ikke nødvendigt at slette maskinen mellem de enkelte opgaver. Hvis De indtaster et ciffer forkert, kan det slettes med **←**, hvorefter De kan indtaste det rigtige ciffer.

Beregning	Indtastning	Lyspanel
$9 - 6 = 3$	9 ENTER 6 -	3,0000
$9 \times 6 = 54$	9 ENTER 6 ×	54,0000
$9 \div 6 = 1,5$	9 ENTER 6 ÷	1,5000
$9^6 = 531.441$	9 ENTER 6 y^x	531.441,0000

Bemærk følgende, der gælder for alle fire eksempler:

- Begge tal er i maskinen, når der trykkes på funktionstasten.
- **ENTER** bruges kun til at adskille to tal, der indtastes lige efter hinanden.
- Resultatet bliver beregnet og udlæst, så snart der er trykket på funktionstasten – i disse tilfælde **-**, **×**, **÷** og **y^x**.

Bemærk: Hvis Deres HP-15C bruger punktum som decimaltegn, kan De selv ændre formatet ved at følge anvisningerne under 3-cifferinddeling (side 60).

For at demonstrere den nære sammenhæng mellem manuel og programmeret problemløsning, vil vi først løse et problem manuelt – dvs. selv foretage alle indtastninger. Bagefter vil vi lave et program, der udfører samme beregning men med andre data.

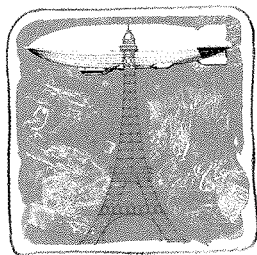
Den tid, det tager et legeme at nå jorden ved et frit fald (vi ser bort fra luftmodstanden), bestemmes af:

$$t = \sqrt{\frac{2h}{g}},$$

hvor t = tiden i sekunder,

h = højden i meter,

g = gravitationskonstanten
(9,8m/sek²).



Eksempel: Beregn den tid, det tager for en sten at nå jorden fra toppen af Eiffeltårnet (300,51 meter højt).

Indtastning

300.51

2

9.8

Lyspanel

300,5100

601,0200

61,3286

7,8313

Indtastning af h .

Beregning af $2h$.

$(2h)/g$.

Faldtid i sekunder.

Programmeret problemløsning

Lad os nu antage, at De skulle beregne faldtiden for forskellige højder. Den simpleste måde at klare dét problem på, er at skrive et program, der udfører de enkelte beregninger, når De har indtastet højden.

Programmet skrives. Programmet svarer til de indtastninger, De netop brugte til den manuelle problemløsning. Det er bekvemt med en etiket i starten af programmet, ligesom det er bekvemt med en returoordre i slutningen. Endelig skal programmet kunne modtage nye data.

Lagring af programmet. De kan lagre et program til løsning af den nævnte opgave ved at trykke på følgende taster i den angivne rækkefølge. (De tal og tegn, der vises i lyspanelet, vil blive forklaret senere.)

Indtastning**Lyspanel**

[g] [P/R]

000 –

Sætter HP-15C i programmeringsstatus og tænder **PRGM** i lyspanelet.

[f] CLEAR [PRGM]

000 –

Sletter programhukommelsen.

[f] [LBL] [A]

001 – 42.21.11

Etiket A angiver starten på programmet.

2

002 –**2**

[×]

003 –**20**

9

004 –**9**

[.]

005 –**48**

8

006 –**8**

[÷]

007 –**10**

[√x]

008 –**11**

[g] [RTN]

009 – 43 32

Samme taster som bruges til den manuelle problemløsning.

Definerer afslutningen på programmet.

[g] [P/R]

7,8313

Skift til normalstatus (hvor **PRGM** er slukket).

Kørsel af programmet. De kan nu køre programmet ved at indtaste faldhøjden og trykke på [A].

Indtastning**Lyspanel**

300.51

300,51

Eiffel-tårnets højde.

[f] [A]

7,8313

Falddid (som De allerede har beregnet én gang).

1050 [f] [A]

14,6385

Falddiden i sekunder for et frit fald over 1.050 meter.

Når programmet er lagret, kan De let beregne falddiden for en vilkårlig højde. De skal blot indtaste højden og trykke [f] [A]. Find falddiden for genstande, der kastes ud fra 100 meters, 2 meters, 275 meters og 2.000 meters højde.

Svarene er: 4,5175 sek, 0,6389 sek, 7,4915 sek og 20,2031 sek.

Dette program var jo nok så enkelt. I kapitel II vil vi gennemgå flere mere komplicerede programmer i nærmere detaljer.

Hvis De ikke er vant til at arbejde med lommeregnere fra Hewlett- Packard, vil vi dog anbefale, at De først læser kapitel I.

Kapitel I
HP-15C
som regnemaskine

Grundbegreberne

Tænd/Sluk-Afbryder

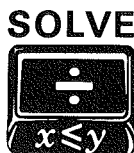
ON bruges til at tænde og slukke for HP-15C.* For at spare på batterierne slukker lommeregneren automatisk, når den ikke har været brugt i nogle få minutter.

Tastfunktioner

Primære og alternative funktioner

De fleste af HP-15C's taster udfører én primær og to alternative funktioner. Den primære funktion for en tast, grundfunktionen, er altid markeret på tastens vandrette overflade. De alternative funktioner er markeret på tastaturpladen lige over tasten og på tastens skrå forside.

- Valg af primærfunktionen, som er trykt på tastens vandrette overflade, udføres ved at trykke direkte på tasten.
- Valg af en alternativ funktion, som er trykt med gul eller blå farve, udføres ved først at trykke på **f** eller **g** og derefter på den tilhørende funktionstast: f.eks. **f** **SOLVE** eller **g** **$x \leq y$** .



I denne brugerhåndbog vil vi følge nogle bestemte konventioner ved reference til alternative funktioner. Vi refererer til *selve funktionen* ved at opgive funktionens navn i en boks, f.eks. " **MEM** -funktionen". Ved reference til *brug af tasten* opgives også den tilhørende forvalgstast, f.eks. "tryk **g** **MEM**". Når vi refererer til de 4 gule funktioner under klammen, der er mærket "CLEAR", vil vi sætte "CLEAR" foran funktionen, f.eks. "CLEAR **REG** -funktionen" eller "tryk **f** CLEAR **PRGM**".

* Bemærk, at **ON** er lavere end de andre taster, så De ikke så let kommer til at tænde for HP-15C ved en fejltagelse.

Bemærk, at når De trykker på **f** eller **g**, vises **f** eller **g** i lyspanelets symbolrække som tegn på, at næste tasttryk aktiverer en alternativ funktion. Indikatoren slukkes først, når De har trykket på en funktionstast.

0.0000

f

Forvalgstaster

En forvalgstast er enhver tast, som skal bruges før en anden tast for at dennes funktion kan udføres. Indtastningen af visse funktioner består af to afsnit: en forvalgstast efterfulgt af et ciffer eller en anden tast. Forvalgstasterne er:

CF	ENG	FIX	GSB	$\frac{\pi}{x}$	MATRIX	SCI	STO
DIM	f	g	HYP	ISG	RCL	SF	TEST
DSE	F?	GTO	HYP ⁻¹	LBL	RESULT	SOLVE	$x \geq$

Hvis De kommer til at bruge en forkert forvalgstast under indtastning, kan De rette fejlen ved at trykke **f** **CLEAR** **PREFIX**. Da **f** **CLEAR** **PREFIX** også bruges til at udlæse alle 10 betydende cifre af tallet i lyspanelregistret, bliver tallet udlæst i ca. 1 sek, når de trykker på **PREFIX**.

Fortegnsskift

Når De trykker på **CHS**, skifter det udlæste tal fortegn. Ved indtastning af et negativt tal skal De først indtaste tallets cifre og derefter trykke på **CHS**.

Indtastning af eksponenter

EEX bruges, når De skal indtaste et tal i flydende tals format, hvor tallet består af en mantisse og en eksponent. De skal indtaste mantissen (de betydende cifre), trykke på **EEX** og derefter indtaste eksponenten (som højst må have to cifre).

Hvis eksponenten er negativ, skal De trykke på **CHS** efter indtastning af eksponenten.* Vi kan f.eks. indtaste Planck's konstant ($6,6262 \times 10^{-34}$ Joule-sek) og gange den med 50:

* **CHS** kan bruges umiddelbart efter **EEX**, men af hensyn til konsistensen i indtastningsformer anbefales det, at De følger den almindelige procedure.

Indtastning**Lyspanel**

6.6262

6,6262**EEX****6,6262 00**

"00" markerer, at de næste cifre, der indtastes, bruges som eksponent.

3

6,6262 03

4

6,6262 34 $(6,6262 \times 10^{34})$.**CHS****6,6262 - 34** $(6,6262 \times 10^{-34})$.**ENTER****6,6262 - 34**

Afslutning af indtastning.

50 **x****3,3131 - 32**

Resultat i Joule-sek.

Bemærk: Decimale cifre, der forsvinder fra lyspanelet, når De trykker på **EEX**, vil blive bevaret i lyspanelregistret.

For at forhindre et forvirrende udlæsningsbillede kan **EEX** ikke bruges sammen med tal, hvor der er flere end 7 cifre foran decimaltegnet eller hvis mantisse er mindre end 0,000001. Hvis De skal indtaste et sådant tal, må De vælge en form med en numerisk større eksponent. F.eks. kan $123456789,8 \times 10^{23}$ indtastes som $1234567,898 \times 10^{25}$ og $0,00000025 \times 10^{-15}$ kan indtastes som $2,5 \times 10^{-22}$.

"CLEAR"-tasterne

Sletning (*clearing* på engelsk) betyder i regnemaskinesammenhæng nulstilling af registre. HP-15C har følgende slettefunktioner (tabellen fortsættes på næste side):

Indtastning	Effekt
g CLx	Nulstiller lyspanelet og lyspanelregistret.
	Sletter sidst indtastede cifre <i>eller</i> lyspanel/lyspanelregistret.
I normalstatus	
I programmeringsstatus	Sletter den udlæste instruktion.
f CLEAR Σ	Nulstiller statistikregistre, lyspanelregistret og rullestakken (beskrives i afsnit 3).

Indtastning	Effekt
f CLEAR PRGM I normalstatus I programme- ringsstatus	Sætter programpilens ved linie 000. Nulstiller hele programhukommelsen.
f CLEAR REG	Nulstiller alle dataregistre.
f CLEAR PREFIX	Ophæver brug af forvalgstast.*
* Udlæser desuden hele mantissen i ca. 1 sek.	

Nulstilling af lyspanel: **CLx** og

HP-15C har to funktioner til at slette eller rette udlæste tal: **CLx** (sletning af X) og  (korrektion af sidst indtastede ciffer).

Normalstatus:

-  **CLx** nulstiller altid lyspanelregistret.
-   sletter det sidst indtastede tal, hvis talindlæsningen ikke er afsluttet (med **ENTER** eller en funktionstast). Med  kan De slette et enkelt ciffer og indtaste det rigtige. Hvis talindlæsningen er afsluttet, vil  fungere præcis som **CLx**.

Indtastning	Lyspanel	
12345	12.345	Talindlæsningen er ikke afsluttet.
	1.234	Sidste ciffer slettes.
9	12.349	
	111,1261	Talindlæsningen afsluttet.
	0,0000	Nu sletter  alle cifre.

Programmeringsstatus:

-  **CLx** er en programmerbar funktion; dvs. den kan lagres i et program, så **g** **CLx** ikke sletter den udlæste instruktion.
-   er *ikke* programmerbar, men vil slette den udlæste instruktion.

Beregninger

Funktioner af én variabel

Funktioner af én variabel virker direkte og alene på tallet i lyspanelregistret. De aktiverer en sådan funktion ved at trykke på den eller de tilhørende funktionstaster, når tallet udlæses.

Indtastning	Lyspanel
45	45
 	1,6532

Funktioner af to variable og

En funktion af to variable kan først udføres, når begge argumenter er lagret i lommeregneren. De aritmetiske operationer , ,  og  er de mest brugte funktioner af to variable.

Afslutning af talindtastning. Hvis De skal indtaste to tal til en funktion af to variable, skal HP-15C have et signal, der viser, at indtastningen af det første tal er afsluttet.  bruges til at *adskille* de to tal. Hvis derimod det første tal allerede er lagret, er der ingen grund til at bruge  inden indtastningen af det andet tal. Alle funktioner, undtagen de der bruges til talindtastning*, afslutter talindtastning.

Bemærk at uanset tallets indtastningsformat vil der altid blive sat decimaltegn og et antal decimaler, når De afslutter talindtastningen (f.eks. med ).

Kæderegning. Bemærk følgende specielle forhold:

-  bruges udelukkende til at adskille tal, der indtastes lige efter hinanden.
- Funktionstasten bruges først, når *begge* tal er indtastet.
- Resultatet af en beregning kan selv bruges som operand. Sådanne mellemresultater lagres og fremkaldes på en "Sidst-Ind, Først-Ud"-basis. Cifre som indtastes efter en funktionsberegning betragtes som taldele i et nyt tal.

* Ciffertasterne, , ,  og .

Eksempel: Beregn $(9 + 17 - 4) \div 4$.

Indtastning	Lyspanel	
9 ENTER	9,0000	Talindtastning afsluttet.
17 +	26,0000	$(9 + 17)$.
4 -	22,0000	$(9 + 17 - 4)$.
4 ÷	5,5000	$(9 + 17 - 4) \div 4$.

Selv endnu mere komplicerede opgaver løses på tilsvarende måde - med automatisk lagring og fremkald af mellemresultater. Det er lettest at starte med de inderste parenteser og arbejde sig ud; på samme måde som ved arbejde med papir og blyant.

Eksempel: Beregn $(6 + 7) \times (9 - 3)$.

Indtastning	Lyspanel	
6 ENTER	6,0000	Beregn først mellemresultatet $(6 + 7)$.
7 +	13,0000	
9 ENTER	9,0000	Beregn dernæst det andet mellemresultat.
3 -	6,0000	
×	78,0000	Til sidst skal de to mellemresultater multipliceres.

Prøv selv at regne nedenstående opgaver. Husk at hver gang De trykker på eller en funktionstast, vil det foregående tal blive lagret og brugt ved udførelsen af næste operation.

$$(16 \times 38) - (13 \times 11) = 465.0000$$

$$4 \times (17 - 12) \div (10 - 5) = 4.0000$$

$$23^2 - (13 \times 9) + 1/7 = 412.1429$$

$$\sqrt{[(5.4 \times 0.8) \div (12.5 - 0.7^2)]} = 0.5998$$

Afsnit 2

Numeriske funktioner

I dette afsnit gennemgår vi HP-15C's numeriske funktioner (med undtagelse af statistik og de avancerede rutiner). De ikke-numeriske funktioner gennemgås særskilt (talindtastning i afsnit 1, stakmanipulation i afsnit 3 og udlæsningsformater i afsnit 5).

Alle funktioner bruges på samme måde, uanset om de udføres manuelt eller under programkontrol. Nogle af funktionerne, som f.eks. **ABS**, har størst interesse i forbindelse med programmering.

Husk at de numeriske funktioner, bortset fra dem der bruges til talindtastning, automatisk afslutter talindtastning. Det betyder, at numeriske funktioner hverken skal følge eller følges af **ENTER**.

Pi

g **π** lagrer de 10 første cifre af π i lyspanelregistret. Det er ikke nødvendigt at bruge **ENTER** til at adskille **π** fra andre tal.

Funktioner til ændring af det udlæste tal

Funktionerne til ændring af det udlæste tal virker på tallet i lyspanelregistret (X-registret).

Heltalsdel. **g** **INT** erstatter tallet i X-registret med det hele tal, der numerisk er mindre end eller lig med tallet.

Decimaldel. **f** **FRAC** erstatter tallet i X-registret med dets decimaldel.

Afrunding. **g** **RND** erstatter tallet i lyspanelregistret med det udlæste tal, som bestemt ved hjælp af **FIX**, **SCI** eller **ENG**.

Numerisk værdi. **g** **ABS** erstatter tallet i lyspanelregistret med dets numeriske værdi.

Indtastning	Lyspanel
123.4567 g INT	123,0000
g LSTx CHS g INT	- 123,0000

Cifrene ændres ikke ved fortegnsskift.

Indtastning

g LSTx f FRAC

1.23456789 CHS

g RND

f CLEAR PREFIX

(slippes)

g ABS

Lyspanel

– 0,4567

– 1,2346

1234600000

– 1,2346

1,2346

Midlertidig udlæsning af alle cifre i mantissen.

Funktioner af én variabel

Funktioner af én variabel påvirker udelukkende tallet i lyspanelregistret (X-registret).

Generelle funktioner

Reciprovkærdi. $\boxed{1/x}$ beregner den reciproke værdi af tallet i X-registret.

Fakultet og Gammafunktionen. $\boxed{x!}$ beregner fakultet af tallet i X-registret, hvis det er et naturligt tal mindre end 70.

$\boxed{x!}$ kan også bruges til beregning af gammafunktionen ($\Gamma(x)$), som indgår i en række matematiske sammenhænge. $\boxed{x!}$ beregner $\Gamma(x + 1)$, så De skal trække 1 fra operanden for at beregne $\Gamma(x)$. Gammafunktionens argumenter er ikke begrænset til de naturlige tal.

Kvadratrod. $\boxed{\sqrt{x}}$ beregner kvadratroden af tallet i X-registret.

Kvadrering. $\boxed{x^2}$ kvadrerer tallet i X-registret.

Indtastning25 $\boxed{1/x}$ 8 $\boxed{x!}$ 3.9 $\boxed{\sqrt{x}}$ 12.3 $\boxed{x^2}$ **Lyspanel**

0,0400

40.320,0000

1,9748

151,2900

8! eller $\Gamma(9)$.

Trigonometriske funktioner

Vinkelmål. De trigonometriske funktioner tager argumenter i det vinkelmål, der er valgt, og som markeres i lyspanelet. Skift af vinkelmål medfører ikke nogen konvertering fra det oprindelige vinkelmål til det nye. Det giver udelukkende en angivelse af, hvilket vinkelmål (grader, radian

eller nygrader) HP-15C skal bruge ved fortolkningen af trigonometriske funktioners argumenter.

[g] [DEG] vælger grader som vinkelmål. Der vises *ingen* indikator i lyspanelet. Med grader menes *decimal*grader ikke sexagesimalgrader (grader/min/sek).

[g] [RAD] vælger radian som vinkelmål. I lyspanelet tændes **RAD** som markering af vinkelmålet. I kompleks-status vil alle funktioner (undtagen **[>R]** og **[>P]**) arbejde med argumenter i radian – uanset hvad der egentlig er valgt som vinkelmål.

[g] [GRD] vælger nygrader som vinkelmål. I lyspanelet tændes **GRAD** som markering af vinkelmålet.

Den kontinuerlige hukommelse vil bevare det sidst valgte vinkelmål, når der slukkes for HP-15C. Første gang der tændes for maskinen efter batteriskift og efter nulstilling af den kontinuerlige hukommelse, vil grader være valgt som vinkelmål.

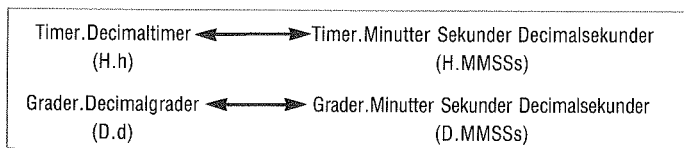
Trigonometriske funktioner. Med x i lyspanelregistret (X-registret) fås:

Indtastning	Resultat
[SIN]	sinus ($\sin x$)
[g] [SIN⁻¹]	Arcus sinus ($\text{Arcsin } x$)
[COS]	cosinus ($\cos x$)
[g] [COS⁻¹]	Arcus cosinus ($\text{Arccos } x$)
[TAN]	tangens ($\tan x$)
[g] [TAN⁻¹]	Arcus tangens ($\text{Arctg } x$)

Før De udfører en trigonometrisk beregning, skal De sikre Dem, at det rette vinkelmål er valgt.

Omregning af tid og vinkler

Tal som repræsenterer tid (timer) og vinkler (grader) kan konverteres af HP-15C således, at der skiftes mellem et decimalformat og et min/sek-format:



Omregning til timer(grader)/min/sek. f ➡H.MS omregner tallet i lyspanelregistret fra decimaltimer(grader) til timer(grader)/minutter/sekunder.

Brug f ➡H.MS til at omregne f.eks.



Brug f PREFIX til at udlæse det størst mulige antal decimale cifre:

1 1 4 0 4 2 0 0 0 0

└──────────┘

Op til 1/100.000-del sekund.

Omregning til decimaltimer(grader). g ➡H omregner tallet i lyspanelregistret fra timer(grader)/minutter/sekunder til decimaltimer(grader).

Omregning mellem grader og radian

g ➡DEG og f ➡RAD bruges til at omregne mellem grader og radian (D.d $\longleftrightarrow$ R.r). Grader skal udtrykkes på decimalform – ikke i sexagesimalform.

Indtastning

40.5 f ➡RAD
g ➡DEG

Lyspanel

0,7069
40,5000

Radian.
 40,5 grader (decimalform).

Logaritmefunktioner

Naturlig logaritme. g LN beregner den naturlige logaritme af tallet i X-registret. Grundtallet er e .

Eksponentialfunktionen. $[e^x]$ beregner eksponentialfunktionen for tallet i X-registret, x – dvs. opløfter e til x 'te potens. Eksponentialfunktionen, e^x , er antilogaritme til den naturlige logaritme.

10-talslogaritme. $[g][\text{LOG}]$ beregner 10-talslogaritme eller den almindelige logaritme af tallet i X-registret, x . Grundtallet er 10.

Almindelig antilogaritme. $[10^x]$ beregner den almindelige antilogaritme af tallet i X-registret, x – dvs. opløfter 10 til x 'te potens.

Indtastning

45 $[g][\text{LN}]$

3.4012 $[e^x]$

12.4578 $[g][\text{LOG}]$

3.1354 $[10^x]$

Lyspanel

3,8067

30,0001

1,0954

1.365,8405

$\ln 45$.

$\exp(3,4012)$ eller $e^{3,4012}$.

$\log 12,4578$.

$10^{3,1354}$.

Hyperbolske funktioner

Med x i X-registret (lyspanelregistret) fås:

Indtastning	Resultat
$[f][\text{HYP}][\text{SIN}]$	hyperbolsk sinus ($\sinh x$)
$[g][\text{HYP}^{-1}][\text{SIN}]$	invers hyperbolsk sinus ($\operatorname{arcsinh} x$)
$[f][\text{HYP}][\text{COS}]$	hyperbolsk cosinus ($\cosh x$)
$[g][\text{HYP}^{-1}][\text{COS}]$	invers hyperbolsk cosinus ($\operatorname{arccosh} x$)
$[f][\text{HYP}][\text{TAN}]$	hyperbolsk tangens ($\tanh x$)
$[g][\text{HYP}^{-1}][\text{TAN}]$	invers hyperbolsk tangens ($\operatorname{arctgh} x$)

Funktioner af to variable

HP-15C beregner alle funktioner af to variable ved at tage to tal, der er indtastet efter hinanden, som argumenter. Hvis De *indtaster* begge tal, skal De huske at adskille dem med $[\text{ENTER}]$ eller en funktion – f.eks. $[g][\text{INT}]$ eller $[1/x]$ – som afslutter talindtastning.

De to variable kaldes henholdsvis x og y , hvor y -værdien er den først indtastede. Den lagres i et register, som kaldes Y-registret. Den anden værdi, x , lagres i X-registret.

De fire regningsarter $+$, $-$, $\times$ og $\div$ er de bedst kendte eksempler på funktioner af to variable. De vil møde andre senere i dette afsnit.

Potensopløftning

y^x opløfter y til den x 'te potens. Grundtallet, y , skal indtastes før eksponenten, x .

Beregning	Indtastning	Lyspanel
$2^{1,4}$	2 [ENTER] 1.4 y^x	2,6390
$2^{-1,4}$	2 [ENTER] 1.4 [CHS] y^x	0,3789
$(-2)^3$	2 [CHS] [ENTER] 3 y^x	- 8,0000
$3\sqrt{2} (2^{1/3})$	2 [ENTER] 3 $1/x$ y^x	1,2599

Procentregning

Procentfunktionerne $\%$ og $\Delta\%$ bevarer det oprindelige grundtal, så det kan bruges til videre beregninger med resultatet.

Procent. $\%$ beregner $x\%$ af y .

De kan f.eks. beregne 3% kontantrabat på en vare, der koster 1.576 kr.

Indtastning

1576 [ENTER]

3 $\%$

$-$

Lyspanel

1.576,0000

47,2800

1.528,7200

Grundtallet (prisen).

Procentbeløbet (kontantrabat).

Nettopris.

Procentisk forskel. $\Delta\%$ beregner den procentiske *forskel* mellem to tal. Resultatet udtrykker stigningen (et positivt resultat) eller faldet (negativt resultat) af x relativt til y .

Vi kan f.eks. betragte en vare, der koster 1.576 kr. mod 1.412 kr. sidste år. Hvor stor er stigningen? Dvs. hvor meget er prisen steget relativt til sidste års pris?

Indtastning

1412 [ENTER]

1576 $\Delta\%$

Lyspanel

1.412,0000

11,6147

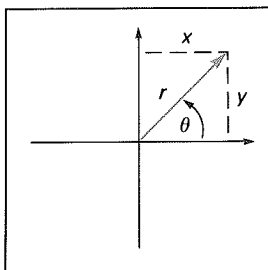
Basistallet (sidste års pris).

Den relative stigning.

Omregning mellem polære og rektangulære koordinater

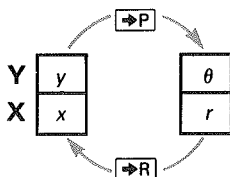
g **→P** og **f** **→R** giver Dem mulighed for at omregne koordinater mellem polære og rektangulære koordinatsystemer.

Vinklen θ regnes for at være givet i det vinkelmål, der markeres i lyspanelet: decimalgrader (ingen direkte markering), radian (**RAD**) eller nygrader (**GRAD**). θ måles som vist på illustrationen til højre.



Omregning til polære koordinater. **g** **→P** omregner et sæt rektangulære koordinater (x, y) til polære koordinater (længde r , vinkel θ). y -værdien skal indtastes før x -værdien. Når De trykker **g** **→P**, bliver r udlæst. Tryk derefter **x $\rightleftharpoons$ y** (*X ombytte med Y*), hvorefter θ føres fra Y-registret til udlæsning i X-registret. θ vil altid ligge i intervallet -180° til 180° , $-\pi$ til π eller -200° til 200° .

Omregning til rektangulære koordinater. **f** **→R** omregner et sæt polære koordinater (længde r , vinkel θ) til rektangulære koordinater (x, y) . θ skal indtastes før r . Når De trykker **f** **→R**, bliver x udlæst. De kan derefter få y udlæst med **x $\rightleftharpoons$ y**.



Indtastning

g **DEG**

5 **ENTER**

10

g **→P**

Lyspanel

5,0000

10

11,1803

Vælger grader som vinkelmål (ingen vinkelindikator).

y -værdi.

x -værdi.

r .

Indtastning
 $x \geq y$

30 ENTER

12

 f $\Rightarrow R$
 $x \geq y$
Lyspanel
26,5651
30,0000
12
10,3923
6,0000
 θ ; rektangulære koordinater
omregnet til polære.

 θ .

 r .

 x -værdi.

 y -værdi; polære koordinater
omregnet til rektangulære.

Afsnit 3

Den automatiske rullestak, SIDSTE X og dataregistre

HP's indtastningssystem er baseret på en matematisk logisk notationsform, som kaldes "Polsk Notation" efter dens opfinder, den polske logiker Jan Łukasiewicz (wuka'fëvitj) (1878-1956). I konventionel algebraisk notation placeres algebraiske operatorer *mellem* operanderne. I Łukasiewicz's notation placeres operatorerne før deres argumenter. For at kunne udnytte systemet bedst muligt i elektronregnersammenhæng har HP i stedet valgt at sætte operatorerne *efter* operanderne. Heraf navnet "Omvendt Polsk Notation" (OPN).

HP-15C bruger OPN til at løse komplicerede udtryk på en fuldstændig ligefrem måde, uden brug af parenteser eller særlige tegn. Det gøres ved automatisk at lagre og fremhente mellemresultater. Dette system er realiseret ved hjælp af den automatiske rullestak og ENTER, som desuden minimerer antallet af indtastninger.

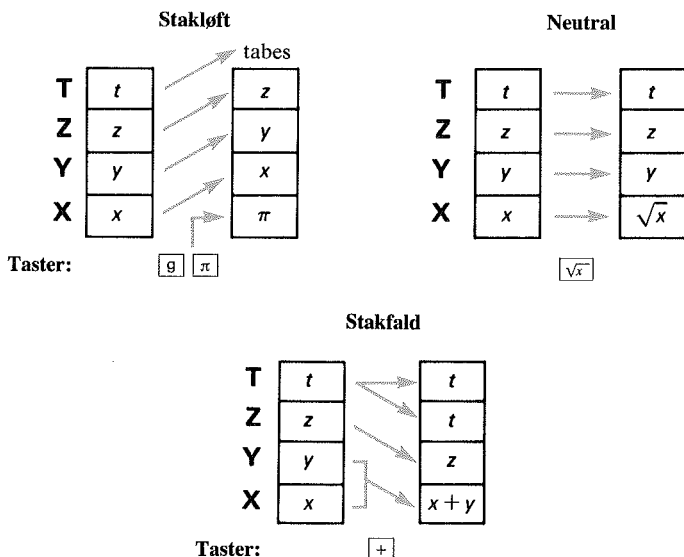
Den automatiske rullestaks registre

T	0.0000	Udlæses altid
Z	0.0000	
Y	0.0000	
X	0.0000	

Når HP-15C er i normalstatus (**PRGM** vises ikke), vil tallet der bliver udlæst altid repræsentere tallet i X-registret.

Ethvert indtastet tal eller resultat af en numerisk funktion lagres i lyspanelregistret (X-registret). Herved vil de tal, som allerede er i rullestakken, enten skifte en plads opad (stakken løftes), blive hvor de er eller skifte en

plads ned (stakken falder) – alt afhængig af de foregående operationer. Tallene i rullestakken lagres på en "sidst-ind, først-ud"-basis. De tre stakke, der er vist nedenfor, illustrerer de tre typer stak-reaktion. x , y , z og t er tal i stakregistrene X, Y, Z og T.

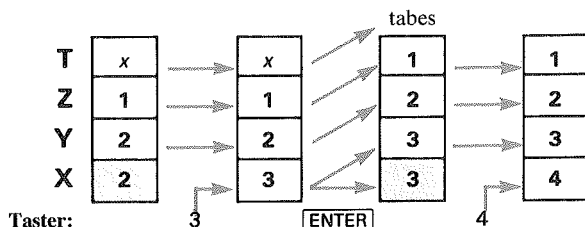
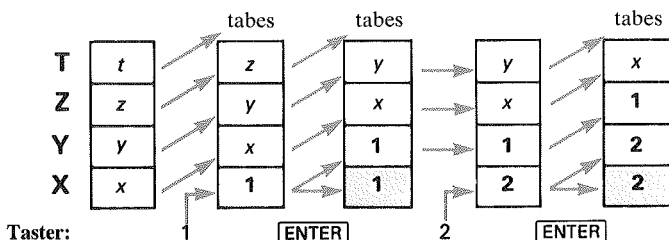


Bemærk, at tallet i T-registret bevares, når stakken falder. De vil senere se, hvorledes dette kan udnyttes til at bruge t som konstant operand.

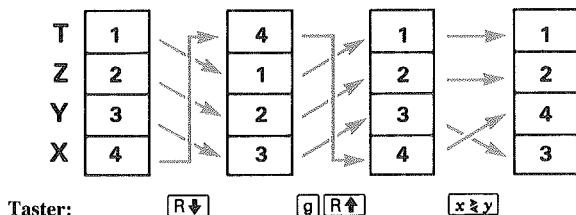
Funktioner til manipulering af stakken

ENTER. Når De trykker på **ENTER** bliver det netop indtastede tal afsluttet, og stakken er klar til at modtage det næste. Klargøringen indebærer, at stakken bliver løftet, og at tallet i X-registret kopieres op i Y-registret. Det næste tal som indtastes vil overskrive indholdet af X-registret; stakken løftes ikke. Eksemplet viser, hvad der sker med stakken, når den fyldes med tallene 1, 2, 3 og 4.

(Der er lagt raster i registre, som overskrives af det næste tal, der indtastes eller fremkaldes – registre, der er i "farezonen").

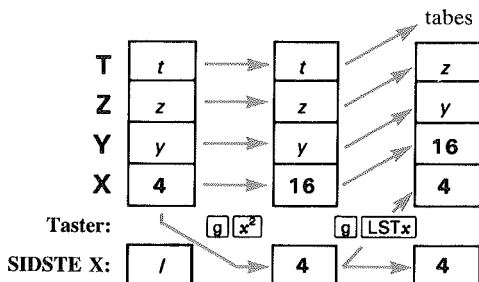


$R\downarrow$ (rul-ned), g $R\uparrow$ (rul-op) og $x \rightleftharpoons y$ (X ombytte med Y). $R\downarrow$ og g $R\uparrow$ bruges til at rotere rullestakken ét register op eller ned uden at der tabs information (dvs. at et tal udveksles mellem X- og T-registrene). $x \rightleftharpoons y$ ombytter tallene i X- og Y-registrene. Hvis rullestakken oprindelig var fyldt med 1, 2, 3 og 4, ville $R\downarrow$, g $R\uparrow$ og $x \rightleftharpoons y$ medføre følgende ændringer:



SIDSTE X-registret og $\boxed{g} \boxed{LSTx}$.

SIDSTE X-registret er et separat register, som indeholder det tal, som var lagret i X-registret umiddelbart før udførelsen af den seneste numeriske funktion.* Når De trykker $\boxed{g} \boxed{LSTx}$ (Sidste X), lagres en kopi af indholdet af SIDSTE X-registret i X-registret. F.eks.:



$\boxed{g} \boxed{LSTx}$ sparer Dem for at genindtaste tal, som De ønsker at bruge igen (denne facilitet omtales nærmere under Konstantaritmetik, side 39). Måske nok så væsentligt er, at den kan bruges til udbedring af indtastningsfejl.

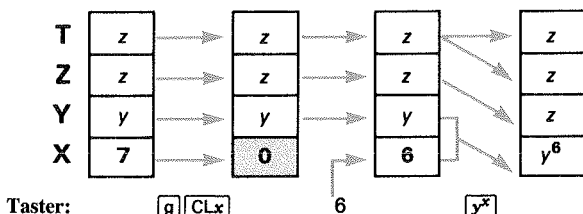
Hvis De f.eks. har indtastet forkert divisor under en kæderegning, kan $\boxed{g} \boxed{LSTx}$ bruges således:

Indtastning287 $\boxed{\text{ENTER}}$ 12.9 $\boxed{\div}$ $\boxed{g} \boxed{LSTx}$ **Lyspanel****287,0000****22,2481****12,9000**

Forkert divisor!

Sidste indhold af X-registret før tryk på $\boxed{\div}$ hentes fra SIDSTE X.

* Medmindre operationen var $\boxed{g} \boxed{\bar{x}}$, $\boxed{g} \boxed{s}$ eller $\boxed{f} \boxed{L.R.}$ som hverken bruger eller bevarer indholdet af X-registret, men i stedet beregner på basis af data, der er lagrede i statistikregistre (R_2 til R_7). I tillæg B kan De se en komplet liste over funktioner, som overfører sidste operand til SIDSTE X-registret.



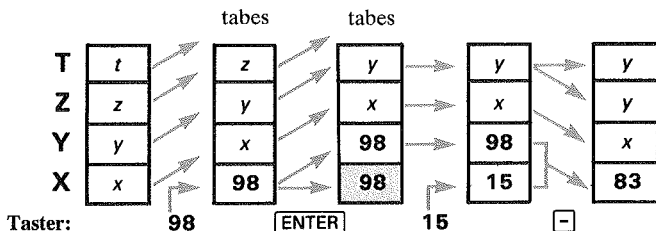
Indtastningsrækkefølge og ENTER

Ved beregning af funktioner af to variable er tallenes placering i rullestakken af allerstørste betydning. Når man skal udføre en aritmetisk funktion, skal tallene lagres i rullestakken på samme måde, som hvis de var skrevet under hinanden på et stykke papir. Man har f.eks.:

$$\begin{array}{r} 98 \\ - 15 \\ \hline \end{array} \quad \begin{array}{r} 98 \\ + 15 \\ \hline \end{array} \quad \begin{array}{r} 98 \\ \times 15 \\ \hline \end{array} \quad \begin{array}{r} 98 \\ 15 \\ \hline \end{array}$$

Som De kan se, skal det første (det øverste) tal lagres i Y-registret, medens det andet skal lagres i X-registret. Når den matematiske operation udføres, falder stakken, og resultatet lagres i X-registret.

F.eks. udføres subtraktion således:



Tallene skal placeres på samme måde ved addition af 98 og 15, ved multiplikation af 98 og 15 og ved division af 98 med 15.

Beregning af udtryk med parenteser

De automatiske løft og fald gør det muligt at beregne udtryk, der af notationsmæssige hensyn er opdelt med parenteser – uden at De skal indtaste parenteser eller lagre mellemresultater i dataregistre. Ethvert aritmetisk udtryk kan beregnes ved hjælp af en sekvens af funktioner af én eller to variable.

Næsten ethvert udtryk kan beregnes med rullestakkens fire registre. Specielt hvis De gør det til en regel altid at starte med det inderste parentes-niveau. Dvs. følge den samme metode, som hvis De skulle udføre beregningen med papir og blyant. Hvis De prøver at udføre beregningen fra venstre mod højre, kan det blive nødvendigt at lagre mellemresultater. Vi vil beregne følgende udtryk:

$$3[4 + 5(6 + 7)]$$

Indtastning

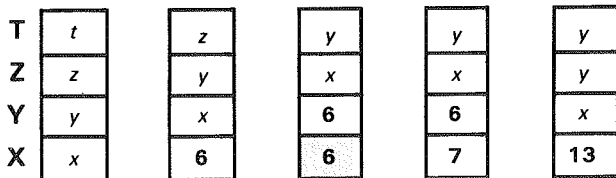
6 7 5 4 3

Lyspanel

13,0000**65,0000****69,0000****207,0000**Mellemresultat: $(6 + 7)$.Mellemresultat: $5(6 + 7)$.Mellemresultat:
 $[4 + 5(6 + 7)]$.

Slutresultat.

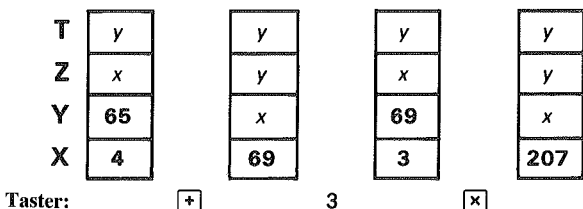
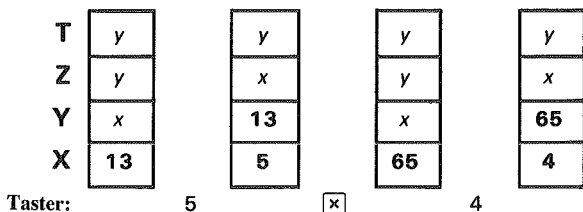
Følgende sekvens illustrerer stakkens bevægelser i dette eksempel. Stakken falder automatisk efter udførelsen af hver funktion af to variable, og bliver løftet, når der igen indtastes et tal. (Af hensyn til overskueligheden vil vi i de resterende eksempler undlade pilene, når vi illustrerer stakbevægelser.)



Taster:

6

7



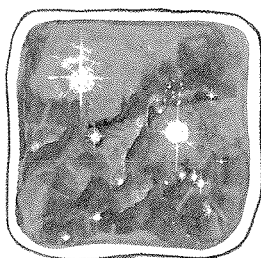
Konstantaritmetik

De kan arbejde med konstantaritmetik på tre måder (uden at bruge dataregistre):

1. Ved brug af SIDSTE X-registret.
2. Ved at fylde rullestakken med en konstant og udføre operationer på *indtastede* tal. (De skal nulstille lyspanelet med g CLx før hver nyindtastning.)
3. Ved at udføre en kæderegning mellem en konstant og indholdet af X-registret.

SIDSTE X. Konstanten skal være lagret i X-registret, når operationen udføres, så den overføres til SIDSTE X-registret. Når De derefter skal bruge konstanten igen, skal De indtaste det variable tal og derefter trykke g LSTx, så konstanten overføres til X-registret.

Eksempel: Blandt de nærmeste stjerner ved Jorden er Rigil Centaurus (4,3 lysår borte) og Sirius (8,7 lysår borte). Brug lysets hastighed, c ($3,0 \times 10^8$ meter/sek eller $9,5 \times 10^{15}$ meter/år), til at bestemme afstanden i meter til disse stjerner. (Stakdiagrammerne viser kun én decimal.)



T	t	z	y	y
Z	z	y	x	x
Y	y	x	4.3	4.3
X	x	4.3	4.3	9.5 15

Taster: 4.3 ENTER 9.5 EEX 15

SIDSTE X:	/	/	/	/
------------------	---	---	---	---

T	y	y	$-y$	x
Z	x	y	x	4.1 16
Y	4.3	x	4.1 16	8.7
X	9.5 15	4.1 16	8.7	9.5 15

Taster: $\times$ 8.7 $\square$ LSTx

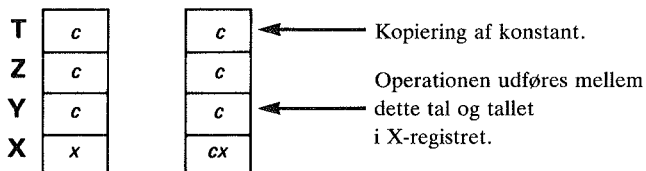
SIDSTE X:	/	9.5 15	9.5 15	9.5 15
------------------	---	--------	--------	--------

T	x	x	
Z	4.1 16	x	
Y	8.7	4.1 16	(Rigil Centaurus er $4,1 \times 10^{16}$ meter borte.)
X	9.5 15	8.3 16	(Sirius er $8,3 \times 10^{16}$ meter borte.)

Taster: $\times$

SIDSTE X:	9.5 15	9.5 15
------------------	--------	--------

Lagring af konstant i alle stakregistre. Da tallet i T-registret bevares, når stakken falder, kan det bruges som konstant i aritmetiske beregninger.



Taster:

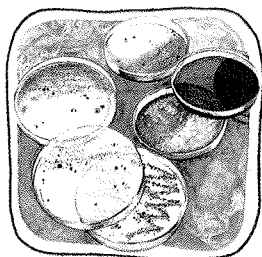


Fyld rullestakken med konstanten ved at indtaste tallet og trykke tre gange på **ENTER**. Indtast første argument og tryk på tasten for den ønskede operation. Stakken falder, indholdet af T-registret kopieres og en kopi af konstanten overføres til Y-registret.

Hvis der skal indtastes et nyt argument ved hver ny konstantberegning, må De huske at slette X-registret med **g** **CLx** inden De indtaster tallet – stakløften er aktiv efter en aritmetisk operation.

Hvis De derimod ikke skal indtaste forskellige argumenter, men arbejde videre med mellemresultatet, skal De ikke slette X-registret, men blot bruge operatortasten det nødvendige antal gange.

Eksempel: En bakteriolog arbejder med en bakteriestamme, hvis størrelse typisk vokser 15% om dagen. Hvis kulturen oprindelig er på 1.000 bakterier, hvor stor er populationen da efter hver dag i fire på hinanden følgende dage?



Indtastning

1.15

ENTER **ENTER**

ENTER

1000

x

Lyspanel

1,15

1,1500

1.000

1.150,0000

Vækstfaktor.

Stakken fyldes.

Startpopulation.

Population efter 1. dag.

Indtastning**Lyspanel**

☐	1.322,5000	Efter 2. dag.
☐	1.520,8750	Efter 3. dag.
☐	1.749,0063	Efter 4. dag.

Brug af dataregistre

Når data lagres eller fremkaldes, bliver de kopieret mellem X-registret (lyspanelregistret) og dataregistre. Når der tændes første gang for HP-15C (dvs. når den kontinuerlige hukommelse er nulstillet), har lommeregneren 21 direkte adresserbare dataregistre: R_0 til R_9 , $R_{.0}$ til $R_{.9}$ samt indeksregistret (R_1) (se registerdiagrammet på omslagets side 3). Seks af registre, R_2 - R_7 , bruges til summation af statistikrutinerne.

Antallet af tilgængelige dataregistre kan forøges eller formindskes. **[DIM]**, som bruges til at bestemme fordelingen af registre (*splitningen*), bliver nærmere omtalt i tillæg C. Ved konvertering af registre bliver dataregistrene med de laveste numre konverteret sidst. *Man bør derfor altid starte fra bunden, når man beslutter, hvilke dataregistre der skal anvendes.*

Lagring og fremkald af tal

[STO] (*store*). Lagrer en kopi af tallet i lyspanelregistret i det dataregister, hvis adresse (0 til 9 eller $.0$ til $.9^*$) indtastes umiddelbart efter **[STO]**. Det tidligere indhold af dataregistret bliver overskrevet.

[RCL] (*recall*). Virker omvendt af **[STO]**, idet tasten udlæser en kopi af det specificerede dataregister. Dataregistrets indhold ændres ikke.

[$\rightleftharpoons$] (*X exchange*). Når tasten efterfølges af en registeradresse (0 til $.9^*$, vil indholdet af lyspanelregistret blive ombyttet med indholdet af det specificerede dataregister. Tasten er specielt anvendelig, når man ønsker at inspicere et dataregister uden at forstyrre rullestakkens indhold.

De ovenstående operationer aktiverer stakløften, så tallet, der bliver stående i X-registret, kan bruges i de efterfølgende beregninger. Hvis De adresserer et register, som ikke er defineret, bliver fejlmeddelelsen **Error 3** udlæst.

* Alle dataregisteroperationer kan desuden udføres med indeksregistret (ved brug af **[I]** eller **[I₁]**). Disse taster omtales i afsnit 10 og i forbindelse med matricer i afsnit 12.

Eksempel: Foråret er på vej, og De vil holde øje med 24 påskeliljer i Deres baghave. De skal lagre antallet af blomstrende liljer den første dag. Til det tal skal De dagen efter addere antallet af nyudsprungne påskeliljer.

Indtastning	Lyspanel	
3 [STO] 0	3,0000	Antallet af udsprungne blomster lagres i R_0 .

Sluk for lommeregneren. Tænd for den næste dag.

Indtastning	Lyspanel	
[RCL] 0	3,0000	Fremkald af antal blomstrende påskeliljer 1. dag.
5 [+]	8,0000	Læg antallet af nyudsprungne liljer til gårdsagens tal.

Nulstilling af dataregistre

Tryk på [f] CLEAR [REG] (*clear registers*) nulstiller *alle* dataregistre, men påvirker hverken rullestakken eller SIDSTE X-registret. De kan nulstille et enkelt dataregister ved at lagre 0 i registret. Når De nulstiller den kontinuerlige hukommelse, bliver alle registre (inklusive stakken) slettet.

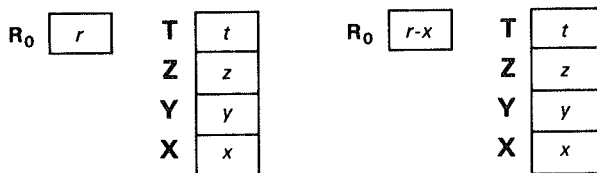
Direkte og omvendt registeraritmetik

Direkte registeraritmetik. Forestil Dem, at De ikke blot vil have et tal lagret, men også vil udføre aritmetiske operationer med tallet og få det erstattet af resultatet af operationen. Dette kan gøres uden brug af [RCL].

1. 1.-operanden (den der svarer til y) skal være lagret i dataregistret, medens 2.-operanden skal være lagret i X.
2. Tryk på [STO].
3. Tryk på [+], [-], [×] eller [÷].
4. Indtast registeradressen (0 til 9, .0 til .9). (Indeksregistret, som omtales i afsnit 10, kan også bruges.)

Efter **direkte registeraritmetik** er det nye tal i registret givet af:

$$\begin{array}{l} \text{nyt indhold} \\ \text{af register} \end{array} = \begin{array}{l} \text{gammelt indhold} \\ \text{af register} \end{array} \left\{ \begin{array}{l} + \\ - \\ \times \\ \div \end{array} \right\} \begin{array}{l} \text{tallet i} \\ \text{lyspanelet} \end{array}$$



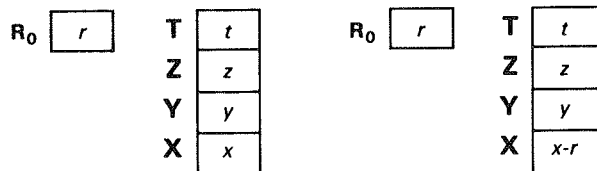
Taster: STO - 0

Omvendt registeraritmetik. Med omvendt registeraritmetik kan De udføre aritmetiske operationer mellem tallet i X-registret og et dataregister, og få lagret resultatet i X-registret uden at stakken løftes. Det betyder, at indholdet af Y, Z og T ikke berøres af operationen.

Omvendt registeraritmetik udføres på samme måde som direkte; De skal blot bruge RCL i stedet for STO.

Efter **omvendt registeraritmetik** er det nye tal i lyspanelregistret givet af:

$$\begin{array}{l} \text{nyt indhold} \\ \text{af X-register} \end{array} = \begin{array}{l} \text{gammelt indhold} \\ \text{af X-register} \end{array} \left\{ \begin{array}{l} + \\ - \\ \times \\ \div \end{array} \right\} \begin{array}{l} \text{tallet i} \\ \text{dataregister} \end{array}$$



Taster: RCL - 0

Eksempel: Hold løbende regnskab med antallet af påskeliljer, der blomstrer.

Indtastning**Lyspanel**8 **[STO]** 0**8,0000**

Antallet af påskeliljer, som blomstrer 2. dag, bliver lagret i R_0 .

4 **[STO]** **[+]** 0**4,0000**

3. dag. Nu blomstrer yderligere 4 liljer.

3 **[STO]** **[+]** 0**3,0000**

4. dag. 3 nyudsprungne planter.

24 **[RCL]** **[-]** 0**9,0000**

Antallet af blomstrende liljer trækkes fra det samlede antal liljer. Der er endnu 9, som ikke har blomstret.

[RCL] 0**15,0000**

Tallet i R_0 ændres ikke af operationen.

Overløb og underløb

Hvis man forsøger at udføre en registeroperation, som vil resultere i overløb i dataregistret, vil tallet i dataregistret blive erstattet af $\pm 9,999999999 \times 10^{99}$ og det udlæste tal vil blinke. Fejltilstanden kan opheves med **[←]**, **[ON]** eller **[g]** **[CF]** 9.

I tilfælde af underløb bliver tallet i dataregistret erstattet af 0.

I dette tilfælde vil det udlæste tal ikke blinke. Over- og underløb omtales nærmere side 60.

Opgaver

1. Beregn værdien af x fra følgende udtryk.

$$x = \sqrt{\frac{8.33(4 - 5.2) \div [(8.33 - 7.46) 0.32]}{4.3(3.15 - 2.75) - (1.71)(2.01)}}$$

Svar: 4,5728

Opgaven kan f.eks. løses således:

4 **[ENTER]** 5,2 **[-]** 8,33 **[×]** **[g]** **[LSTx]** 7,46 **[-]** 0,32 **[×]** **[÷]** 3,15 **[ENTER]**
 2,75 **[-]** 4,3 **[×]** 1,71 **[ENTER]** 2,01 **[×]** **[-]** **[÷]** **[√x]**

2. Brug registeraritmetik til at beregne restgælden af et lån på 1.000 kr. efter 6 ydelser à 100 kr, når rentesatsen er 1% pr. termin (betalingsperiode).

Procedure: $(1 + i)$ lagres i stakken (i er rentesatsen som decimal) hvorefter hovedstolen (lånebeløbet) indtastes. Følgende formel bruges til at beregne restgælden efter hver ydelse:

$$\text{Ny restgæld} = ((\text{Gammel restgæld}) \times (1 + i)) - \text{Ydelse}$$

Første del af tastsekvensen bliver:

1.01 1000

Indtast

100

for hver ydelse.

Restgæld efter 6 ydelser: 446,32 kr.

3. Gem 100 i R_5 .

1. Dividér indholdet af R_5 med 25.
2. Træk 2 fra indholdet af R_5 .
3. Gang indholdet af R_5 med 0,75.
4. Læg 1,75 til indholdet af R_5 .
5. Fremkald indholdet af R_5 .

Svar: 3,2500.

Statistiske funktioner

Inden De går i gang med dette afsnit bør De sikre Dem, at De virkelig har forstået den automatiske rullestaks virkemåde. Indtastningsrækkefølgen er af afgørende betydning i forbindelse med de fleste statistiske beregninger.

Sandsynlighedsregning

Ved beregninger med permutationer og kombinationer kan der udelukkende regnes med *naturlige tal*. y -værdien skal indtastes før x -værdien. Disse funktioner vil, ligesom de aritmetiske, få stakken til at falde, når resultatet lagres i X.

Permutationer. $\boxed{f} \boxed{P_{y,x}}$ beregner antallet af mulige *sorterede arrangementer* af y forskellige genstande taget x ad gangen. Ingen genstand optræder mere end én gang i et arrangement. Arrangementer af samme genstande i forskellig orden tælles med. F.eks. regnes ABC og ACB for forskellige arrangementer. Basisformlen er:

$$P_{y,x} = \frac{y!}{(y-x)!}$$

Kombinationer. $\boxed{g} \boxed{C_{y,x}}$ beregner antallet af *usorterede arrangementer* af y forskellige genstande taget x ad gangen. Ingen genstand optræder mere end én gang i et arrangement. Arrangementer af samme genstande i forskellig orden tælles *ikke* med. F.eks. regnes ABC og ACB for ens arrangementer. Basisformlen er:

$$C_{y,x} = \frac{y!}{x!(y-x)!}$$

Eksempler: Hvor mange forskellige usorterede arrangementer kan der laves af fem malerier, som skal hænges op på en væg tre ad gangen?

Indtastning	Lyspanel	
5 [ENTER] 3	3	5 (y) billeder hængt op 3 (x) ad gangen.
[f] [Py,x]	60,0000	Der er 60 forskellige ophængninger.

Hvor mange forskellige hænder à 4 kort kan der gives fra et spil med 52 kort?

Indtastning	Lyspanel	
52 [ENTER] 4	4	52 (y) kort givet 4 (x) ad gangen.
[g] [Cy,x]	270.725,0000	Antal forskellige hænder.

Under beregningen der, afhængig af tallenes størrelse, kan være adskillige sekunder, vises **running**. x og y må hver for sig ikke være større end 9.999.999.999.

Pseudotilfældige tal

[f] [RAN#] (*random*) udlæser et pseudotilfældigt tal (fra en sekvens af jævnt fordelte, pseudotilfældige tal) i intervallet $0 \leq r < 1$.*

Når der første gang tændes for HP-15C (og efter nulstilling af den kontinuerlige hukommelse) bliver tallet 0 brugt som udgangstal af den generator, der danner sekvensen. Hver gang De får udlæst et tilfældigt tal vil det blive brugt som udgangstal for det næste tal. De kan initialisere en ny sekvens ved at lagre et nyt udgangstal. (Bemærk at samme udgangstal giver samme sekvens.)

[STO] [f] [RAN#] lagrer det udlæste tal ($0 \leq r < 1$) som nyt udgangstal for sekvensen af jævnt fordelte pseudotilfældige tal. Hvis De indtaster en r -værdi uden for det specificerede interval, vil HP-15C selv normalisere tallet.

[RCL] [f] [RAN#] fremkalder (udlæser) det aktuelle udgangstal.

* Består spektraltesten (D.Knuth, *The Art of Computer Programming*, vol. 2, 1969).

Indtastning**Lyspanel**.5764 [STO] [f] [RAN#] **0,5764**

0,5764 lagres som udgangstal. ([f] kan undværes.)

[f] [RAN#] **0,3422**

1. tal i sekvensen.

[f] [RAN#] **0,2809**

2. tal i sekvensen.

[←] **0,0000**[RCL] [f] [RAN#] **0,2809**

Fremkald af det sidst genererede tal (nyt udgangstal). ([f] kan udelades.)

Summation af statistikdata

HP-15C kan udføre statistiske beregninger med én- og to-dimensionale datasæt. Datapar indtastes i Y- og X-registrene, hvorefter [Σ+] bruges til automatisk bearbejdning og summation af de indtastede data. Summerne lagres i dataregistrene R₂ til R₇. Disse registre omtales derfor ofte som *statistikregistre*.

Før De begynder at summere et datasæt, skal De bruge [f] CLEAR [Σ] til at nulstille statistikregistre og rullestakken. (Hvis De har splittet hukommelsen således, at nogle eller alle statistikregistre er blevet konverteret til programregistre, vil **Error 3** blive udlæst, når De forsøger at benytte CLEAR [Σ], [Σ+] eller [Σ-]. I tillæg C kan De læse, hvordan man ændrer splitningen.)

Ved beregninger på én-dimensionale datasæt skal De efter indtastning af den enkelte x -værdi trykke på [Σ+].

Ved beregninger på to-dimensionale datasæt skal De indtaste de enkelte datasæt (x - og y -værdierne) således:

1. Indtast først y -værdien.
2. Tryk på [ENTER]. Den udlæste y -værdi bliver kopieret op i Y.
3. Indtast x .
4. Tryk på [Σ+]. Det aktuelle antal datapar, n , bliver udlæst. x -værdien bliver lagret i SIDSTE X-registret og y forbliver i Y-registret. [Σ+] blokerer stakløften, så stakken ændres ikke, når næste tal bliver indtastet.

I nogle tilfælde, hvor forskellen på de givne x - og/eller y -værdier er relativt lille, kan HP-15C ikke beregne s , r , lineær regressionskoefficienter og $\hat{y}$,

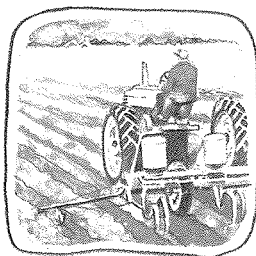
og vil udlæse fejlmeddelelsen **Error 2**. Dette vil dog ikke ske, hvis De normaliserer datasættet således, at summerne dannes af differenserne mellem de enkelte værdier og middelværdien eller en tilnærmelse til middelværdien. Efter beregning af $\bar{x}$, $\bar{y}$ og skæringspunkt ($\boxed{f}$ $\boxed{L.R.}$) skal resultaterne tilbageskaleres. Hvis f.eks. x -værdierne er 665999, 666000 og 666001, bør De betragte disse data som -1, 0 og 1 for senere at addere 666000 til de relevante data.

Der arbejdes med følgende summer:

Register	Indhold	
R ₂	n	Antallet af summerede data(par) (n bliver desuden udlæst).
R ₃	Σx	Sum af x -værdier (x_i).
R ₄	Σx^2	Kvadratsum af x -værdier.
R ₅	Σy	Sum af y -værdier (y_i).
R ₆	Σy^2	Kvadratsum af y -værdier.
R ₇	Σxy	Produktsum af x - og y -værdier.

De kan udlæse enhver af de beregnede summer ved at trykke på $\boxed{RCL}$ og nummeret på det dataregister, som indeholder den ønskede sum. Hvis De bruger $\boxed{RCL}$ $\boxed{\Sigma+}$, vil Σx og Σy blive kopieret over til henholdsvis X og Y . Tastsekvensen $\boxed{RCL}$ $\boxed{\Sigma+}$ løfter stakken to gange, hvis stakløften er aktiv. Hvis stakken er blokeret eller neutral, vil stakken blive løftet én gang og efterlades aktiv.

Eksempel: Agronomen Jørgen Bonde har udviklet en ny type hvede, og han har målt høstudbyttet som en funktion af forbrugt kunstgødning. Brug $\boxed{\Sigma+}$ til at summere de nedenstående data og find værdierne for Σx , Σx^2 , Σy , Σy^2 og Σxy for kvælstofforbruget (x) og høstudbyttet (y).



X	N-gødning kg/hektar	0,00	20,00	40,00	60,00	80,00
Y	Høstudbytte tons/hektar	4,63	4,78	6,61	7,21	7,78

Indtastning**Lyspanel**f CLEAR Σ **0,0000**

Nulstilling af statistikregistre (R_2 til R_7) samt stakken.

f FIX 2

0,00

Valg af samme antal decimaler som i tabellen.

4.63 ENTER

4,630 $\Sigma+$ **1,00**

Første datapar.

4.78 ENTER

4,7820 $\Sigma+$ **2,00**

Andet datapar.

6.61 ENTER

6,6140 $\Sigma+$ **3,00**

Tredie datapar.

7.21 ENTER

7,2160 $\Sigma+$ **4,00**

Fjerde datapar.

7.78 ENTER

7,7880 $\Sigma+$ **5,00**

Femte datapar.

RCL 3

200,00

Sum af x -værdier, Σx (kg kvælstofgødning).

RCL 4

12.000,00

Kvadratsum af x -værdier, Σx^2 .

RCL 5

31,01

Sum af y -værdier, Σy (høstudbytte).

RCL 6

200,49

Kvadratsum af y -værdier, Σy^2 .

RCL 7

1.415,00

Produktsum af x - og y -værdier, Σxy .

Korrektion af summerede data og datapar

Hvis De opdager, at De har fejlintastet data, kan fejlen let rettes.

Ved korrektion af én-dimensionale datasæt skal De blot genindtaste det

forkerte tal og trykke på $\boxed{g} \boxed{\Sigma-}$, hvorefter tallet er taget ud af materialet.

Ved korrektion af to-dimensionale datasæt skal både x - og y -værdien rettes, selv om måske kun den ene er forkert.

1. Genindtast det fejlindtastede datasæt.
2. Brug $\boxed{g} \boxed{\Sigma-}$ til at slette datasættet.
3. Indtast de korrekte x - og y -værdier.
4. Tryk på $\boxed{\Sigma+}$.

Hvis De skal slette det sidst indtastede tal eller datapar (og *har* trykket på $\boxed{\Sigma+}$), kan De nøjes med at bruge $\boxed{g} \boxed{LSTx} \boxed{g} \boxed{\Sigma-}$.*

Eksempel: Efter indtastning af de foregående data bliver Bonde klar over, at han har fejllæst et tal i sine optegnelser. Den anden y -værdi skulle have været 5,78 i stedet for 4,78. Ret denne fejl.

Indtastning	Lyspanel	
4.78 $\boxed{ENTER}$	4,78	Indtastning af det
20 $\boxed{g} \boxed{\Sigma-}$	4,00	forkerte datapar. Bemærk, at
		n skifter til 4.
5.78 $\boxed{ENTER}$	5,78	Det rigtige talpar
20 $\boxed{\Sigma+}$	5,00	indtastes, og n -værdien skif-
		ter tilbage til 5.

Vi vil bruge disse tal senere, så De bør ikke slette tallene.

Middelværdi

$\boxed{g} \boxed{\bar{x}}$ beregner aritmetisk middelværdi af x - og y -værdier efter formlerne i tillæg A. Når De bruger $\boxed{g} \boxed{\bar{x}}$, bliver stakken løftet (to gange hvis stakløften er aktiv, én gang hvis den er blokeret eller neutral); middelvær-

* Bemærk at denne rettellesmetode ikke eliminerer de afrundingsfejl, der allerede er opstået under summationen af det eller de tal, der bliver slettet. De kan komme ud for tilfælde, hvor alle data er slettet og statistikregistrene alligevel ikke er nulstillede. Normalt får afrundingsfejlene dog ingen betydning. Men hvis det fejlindtastede tal/talpar er størrelsesordener større end de øvrige data, bør De slette statistikregistrene og starte forfra.

dien af x_i ($\bar{x}$) lagres i X, medens middelværdien af y_i ($\bar{y}$) samtidig lagres i Y. De kan bruge $x \geq y$ til at udlæse $\bar{y}$.

Eksempel: Beregn den gennemsnitlige mængde gødning ($\bar{x}$) og det gennemsnitlige høstudbytte ($\bar{y}$) af de rettede data fra det foregående eksempel.

Indtastning	Lyspanel	
$\bar{g}$ $\bar{x}$	40,00	Middelværdi for mængden af anvendt kvælstofgødning, $\bar{x}$.
$x \geq y$	6,40	Middelværdi for høstudbyttet, $\bar{y}$.

Standardafvigelse

$\bar{g}$ s beregner *standardafvigelsen* for de summerede data. Formlerne der bliver brugt til at beregne s_x , standardafvigelsen for x_i , og s_y , standardafvigelsen for y_i , kan findes i tillæg A.

Denne funktion giver et estimat af populationsstandardafvigelsen på basis af den givne stikprøve, og kaldes derfor også stikprøvestandardafvigelsen.* Når De bruger $\bar{g}$ s , bliver stakken løftet (to gange hvis stakløften er aktiv, én gang hvis den er blokeret eller neutral); s_x lagres i X, medens s_y samtidig lagres i Y. De kan bruge $x \geq y$ til at udlæse s_y .

Eksempel: Bestem standardafvigelserne for stikprøverne.

Indtastning	Lyspanel	
$\bar{g}$ s	31,62	Standardafvigelsen for gødningsmængden, s_x .
$x \geq y$	1,24	Standardafvigelsen for høstudbyttet, s_y .

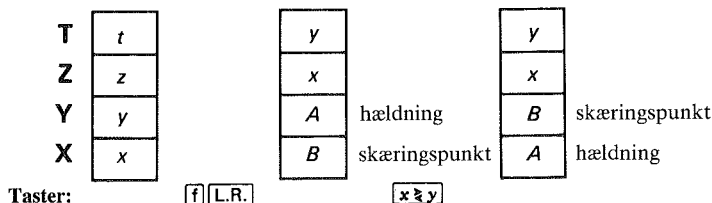
* Når Deres data består af hele populationen og ikke blot en stikprøve, vil den beregnede standardafvigelse være den sande populationsstandardafvigelse, σ . Formlen for den sande populationsstandardafvigelse skiller sig fra stikprøvestandardafvigelsen med en faktor $\sqrt{(n-1)/n}$. For store værdier af n er forskellen negligerbar. Men De kan beregne populationsstandardafvigelsen ved at inddrage middelværdien i datasættet inden De beregner standardafvigelsen. De skal altså trykke $\bar{g}$ $\bar{x}$ $\Sigma+$ $\bar{g}$ s . Men husk at trække middelværdien ud af datasættet, hvis De skal arbejde videre med det.

Lineær regression

Lineær regression er en statistisk metode til at finde den rette linie, der (efter mindste kvadraters metode) passer bedst til to eller flere datapar. Man taler om en *fitning* af et datasæt. Metoden bruges bl.a. til at vurdere, om der eksisterer en lineær sammenhæng mellem de givne data eller en given transformation. $\boxed{f}$ $\boxed{L.R.}$ bestemmer *regressionsliniens* hældningskoefficient, A , og skæringspunkt på y -aksen, B :

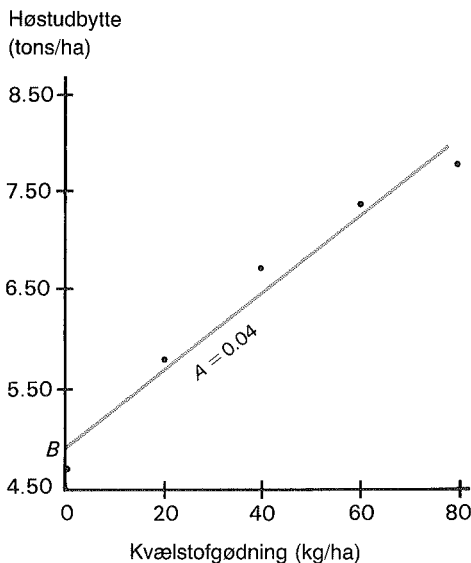
$$y = Ax + B$$

1. Summér data med $\boxed{\Sigma+}$
2. Brug $\boxed{f}$ $\boxed{L.R.}$. Regressionsliniens skæringspunkt på y -aksen, B , bliver udlæst (lagres i X-registret). Hældningskoefficienten, A , lagres samtidig i Y-registret.
3. Tryk på $\boxed{x \geq y}$, når A skal udlæses. (Også $\boxed{f}$ $\boxed{L.R.}$ vil løfte stakken to gange, hvis den er aktiv, og ellers kun én gang.)



Regressionsliniens skæringspunkt og hældningskoefficient bliver beregnet ud fra formler, der vises i tillæg A.

Eksempel: Find y -skæringspunkt og hældningskoefficient for den lineære tilnærmelse for de givne data og sammenlign med nedenstående graf.



Indtastning

Lyspanel

4,86

0,04

Skæringspunkt.

Hældningskoefficient.

Lineære estimater og korrelationskoefficienter

Når De bruger , bliver det *lineære estimat*, $\hat{y}$, beregnet og udlæst. Samtidig beregnes *korrelationskoefficienten*, r , som lagres i Y. r udlæses med .

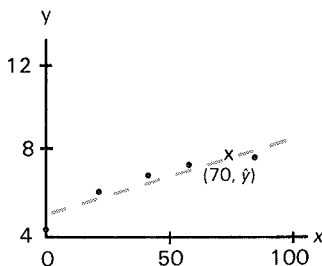
Lineært estimat. Det lineære estimat er den y -værdi på regressionslinien, som svarer til den indtastede x -værdi.

Korrelationskoefficient. Den lineære regression har kun mening, hvis der eksisterer en rimeligt snæver (talmæssigt) lineær sammenhæng mellem de givne x - og y -værdier – en korrelation. Korrelationskoefficienten, r , er et mål for hvor godt den rette linie passer til de givne data.

r kan antage værdier i intervallet $-1 \leq r \leq 1$, hvor -1 repræsenterer den perfekte negative korrelation medens $+1$ repræsenterer den perfekte positive korrelation.

Bemærk, at hvis De ikke indtaster en x -værdi før De trykker på $\boxed{f}$ $\boxed{\hat{y}, r}$, vil X-registrets forrige indhold blive brugt til beregning af $\hat{y}$.

Eksempel: Hvor stort bliver udbyttet, hvis man bruger 70 kg/ha til hvedemarken? Brug de givne data til at estimere udbyttet, og udlæs bagefter korrelationskoefficienten for at se, om det overhovedet giver mening at bruge Bondes data til lineær estimering.



Indtastning

70 $\boxed{f}$ $\boxed{\hat{y}, r}$
 $\boxed{x \geq y}$

Lyspanel

7,56
0,99

Estimeret udbytte.
 Korrelationen er fornem, så
 vi tror på tallet.

Andre anvendelser

Interpolation. De kan bruge $\boxed{\hat{y}, r}$ til lineær interpolation i tabeller; lineær interpolation og estimering er samme sag. Man går i begge tilfælde ud fra, at to konsekutive tabelværdier udgør to punkter på en ret linie, og at den ukendte funktionsværdi ligger på denne rette linie.

Vektoraritmetik. Summationsregistrene kan også bruges til at udføre addition og subtraktion af vektorer. Vektorer på polær form skal først omregnes til rektangulært format (med θ $\boxed{\text{ENTER}}$ r $\boxed{f}$ $\boxed{\rightarrow R}$ $\boxed{\Sigma+}$). Resultatet fremkaldes fra R_3 (Σx) og R_5 (Σy) (med $\boxed{\text{RCL}}$ $\boxed{\Sigma+}$) og omregnes om nødvendigt til polære koordinater.

Efter indtastning af den anden vektor skal der trykkes på $\boxed{\Sigma+}$, hvis vektoren skal adderes, og $\boxed{g}$ $\boxed{\Sigma-}$ hvis den skal trækkes fra.

Afsnit 5

Lyspanelet og den kontinuerlige hukommelse

Udlæsningsformater

HP-15C har tre udlæsningsformater – **FIX**, **SCI** og **ENG** – som bruger et én-cifret tal til nærmere at specificere formatet. Nedenfor vises, hvordan tallet 123.456 bliver udlæst, når man bruger 4 til at specificere formatet:

f	FIX	4	:	123.456,0000
f	SCI	4	:	1,2346 05
f	ENG	4	:	123,46 03

Takket være den kontinuerlige hukommelse vil enhver ændring blive bevaret indtil den kontinuerlige hukommelse nulstilles (eller formatet ændres igen).

Det valgte format aktiveres efter indtastning; under selve indtastningen vises indtil 10 betydende cifre uanset formatet.

Fastkomma

FIX *n* (*fixed decimal*) udlæser tal afrundet til *n* decimaler. *n* kan være et heltal mellem 0 og 9. HP-15C udlæser med prioritering af de mest betydende cifre, så antallet af decimaler er højst 10-*i*, hvor *i* er antallet af heltalscifre. Formatet skifter til flydende tal, hvis tallet er enten for lille eller for stort til at blive udlæst med fastkomma. Normalformatet for HP-15C (formatet, når der første gang tændes) er **FIX** 4.

Indtastning

123,4567895

f **FIX** 4

f **FIX** 6

f **FIX** 4

Lyspanel

123,4567895

123,4568

123,456790

123,4568

Tallet udlæses afrundet til 6 decimaler (internt er der lagret 10).

Sædvanlige **FIX** 4-format.

Flydende tals repræsentation

f **SCI** n (*scientific notation*) udlæser tal med en normaliseret mantisse (dvs. med ét ciffer foran decimaltegnet) og n cifre efter decimaltegnet, så der i alt udlæses n cifre. Der kan udlæses indtil 6 cifre efter decimaltegnet. Den resterende plads optages af eksponenten og dennes fortegn. Men tallet bliver altid afrundet til det valgte antal cifre, så hvis De specificerer flere cifre end lyspanelet kan vise - dvs. **f** **SCI** 7, 8 eller 9, vil afrundingen finde sted ved det *skjulte* syvende, ottende eller niende ciffer efter decimaltegnet.*

Med det foregående tal fås:

Indtastning	Lyspanel	
f SCI 6	1,234568 02	Afrunding til 6. ciffer efter decimaltegnet.
f SCI 8	1,234567 02	Afrunding til det 8. ciffer efter decimaltegnet.

Metrisk notation

f **ENG** n udlæser også tal med flydende notation, blot er mantissen ikke normaliseret. I stedet holdes eksponenten altid som et multiplum af 3, og der udlæses i alt (n) cifre, hvor n højst kan være 6.

Indtastning	Lyspanel	
.012345	0,012345	
f ENG 1	12, - 03	Afrunder til det første ciffer efter startcifret.
f ENG 3	12,35 - 03	
10 x	123,5 - 03	Decimaltegnet flyttes, så eksponenten bevares som et multiplum af 3.
f FIX 4	0,1235	Sædvanlige FIX 4-format.

* Derfor vil lyspanelet udlæse samme tal for **SCI** 7, 8 og 9 - medmindre det afrundede ciffer er 9, som fører 1 videre til nærmeste ciffer.

Udlæsning af hele mantissen

Uanset udlæsningsformatet vil HP-15C altid internt arbejde med en 10-cifret mantisse og en 2-cifret eksponent. F.eks. vil π altid være repræsenteret ved $3,141592654 \times 10^{00}$, uanset hvad der udlæses.

De kan få hele mantissen i X-registret udlæst ved at trykke f CLEAR PREFIX. Tallet udlæses så længe tasten holdes nede.

Indtastning

g π

f CLEAR PREFIX

(hold)

Lyspanel

3,1416

3141592654

Afrundingsfejl

Som tidligere omtalt vil HP-15C repræsentere ethvert tal med en 10-cifret mantisse. Det betyder specielt, at resultater afrundes til 10 cifre. Da maskinen kun kan arbejde med en endelig tilnærmelse til tal som f.eks. π eller $2/3$ (0,666...), kan der blive tale om en ringe afrundingsfejl. Denne fejl kan akkumuleres i mere omfattende beregninger, men normalt er den betydningsløs. I *HP-15C Advanced Functions Handbook* kan De finde en mere detaljeret diskussion af disse forhold.

Lyspanelet som indikatorpanel

Symbolrækken

HP-15C har en symbolrække med otte indikatorer i lyspanelet. Disse indikatorer angiver maskinens tilstand under forskellige operationer. På de følgende sider vil meningen med og brugen af disse indikatorer blive gennemgået.

*	Batteriindikator, side 61.
USER	USER-status, siderne 78 og 142.
f og g	Forvalgstaster, siderne 18-19.
RAD og GRAD	Vinkelmål, side 25.
C	Kompleksstatus, side 119.
PRGM	Programmeringsstatus, side 64.

3-cifferinddeling

Når der tændes for HP-15C, er der valgt et standardformat, hvor punktum bruges som decimaltegn og komma til 3-cifferinddeling af cifrene foran decimaltegnet. Som allerede nævnt kan De selv ændre dette format, så komma bliver brugt som decimaltegn: Tryk på **ON**, hold tasten nede medens De trykker på **.** Slip først **ON** og dernæst **.** (**ON** / **.**). (Hvis De gentager sekvensen, skifter HP-15C tilbage til det oprindelige format.)

Indtastning

12345.67

ON / **.****ON** / **.**

Lyspanel

12.345,67**12,345.6700****12.345,6700**

(Vi går ud fra, at
De allerede har
skiftet om...)

Fejlmeddelelser

Hvis De forsøger at udføre en ugyldig operation - som f.eks. division med nul - vil der blive udlæst en fejlmeddelelse på formen **Error n**, hvor **n** er et ciffer. I tillæg A kan De finde en fuldstændig liste over fejlmeddelelser og deres årsag.

Overløb og underløb

Overløb. Hvis tallet i et vilkårligt register bliver numerisk større end $9,999999999 \times 10^{99}$, bliver $\pm 9,999999999 \times 10^{99}$ lagret i det pågældende register og overløbsflaget, flag 9, bliver samtidig sat.* Flag 9 får det udlæste tal til at blinke. Når overløb indtræffer i et kørende program, vil udførelsen fortsætte indtil programmet er færdigt, hvorefter resultatet udlæses blinkende.

Den blinkende udlæsning kan ophæves med **◀**, **ON** eller **g** **CF** 9.

Underløb. Hvis tallet i et vilkårligt register bliver numerisk mindre end $1,000000000 \times 10^{-99}$, bliver det erstattet af nul. Underløb påvirker ikke maskinen på nogen anden måde.

* Husk at mantissens 3 sidste cifre ikke bliver udlæst.

Batteriindikator

Når en stjerne begynder at blinke i nederste venstre side af lyspanelet, er det tegn på, at batterierne skal udskiftes; men der er ingen grund til panik. Fra stjernen begynder at blinke, er der energi til mindst 10 minutters programkørsel og mindst én times manuel brug. I tillæg F beskrives alle forhold omkring udskiftning af batterier.

0.0000

Kontinuerlig hukommelse

Status

Den kontinuerlige hukommelse bevarer al nødvendig information om følgende tilstande og formater, når der er slukket for lommeregneren:

- Alle lagrede numeriske data.
- Alle lagrede programmer.
- Programpilens placering.
- Udlæsningsformat og antal decimaler.
- Vinkelmål.
- Returadresser for kaldte underprogrammer.
- Flagenes status (med undtagelse af flag 9, som slettes, når lyspanelet afbrydes *manuelt*).
- USER-status.
- Komplexstatus.

Når HP-15C tændes første gang (efter nulstilling af den kontinuerlige hukommelse) er den i normaltilstand (RUN-status).

Hvis De i forvejen har slukket manuelt for HP-15C, vil den kontinuerlige hukommelse bevare ovennævnte information, selv om De fjerner batterierne et øjeblik. Data og programmer bevares i længere tid end statusinformation. Se tillæg F, hvor batteriskift omtales i nærmere detaljer.

Nulstilling af den kontinuerlige hukommelse

Hvis De ønsker at nulstille alle registre og normalstille alle tilstandsfunktioner, skal De udføre følgende proces:

1. Sluk for HP-15C.
2. Tryk på **ON** og hold tasten nedtrykket medens De trykker på **-**.
3. Slip først **ON** og dernæst **-**. (Denne proces vil vi symbolisere med **ON** / **-**.)

Når den kontinuerlige hukommelse er nulstillet, bliver **Pr Error** (*power error*) udlæst. Systemmeddelelsen slettes, så snart De trykker på en vilkårlig tast.

Bemærk: Den kontinuerlige hukommelse kan blive nulstillet, hvis De taber HP-15C på gulvet eller på anden måde udsætter den for overlast.

Kapitel II
HP-15C
Programmering

Afsnit 6

Grundlæggende programmering

De næste fem afsnit bruges alene til at forklare visse grundlæggende programmeringsbegreber. I hvert afsnit vil vi først gennemgå de helt basale metoder (Mekanikken), hvorefter vi ser på nogle eksempler på anvendelsen af disse metoder (Eksempler) og endelig vil vi diskutere mere indviklede punkter i nærmere detaljer (Yderligere information). De behøver kun læse De afsnit, som De selv synes har værdi for Dem.

Mekanikken

Sådan skrives et program

Det er meget let at programmere HP-15C; udgangspunktet er de samme indtastninger, som bruges ved manuel løsning af opgaven. Af ekstraarbejde kommer kun, at De skal beslutte hvor og hvordan data skal indtastes, programmet skal indtastes og lagres og der skal indføres nogle få styreoperationer. Men dertil kommer, at De kan lave programmer, som selv afgør det videre programforløb på basis af de opnåede resultater - som arbejder med forgreninger, sammenligningsoperationer, løkker osv.

Sådan lagres et program

Programmeringsstatus.   (*program/run*) sætter HP-15C i programmeringsstatus, hvor **PRGM** vises i lyspanelet, og hvor indtastninger lagres som programinstruktioner i stedet for at blive udført.

Indtastning

Lyspanel

000 -

Skift til programmeringsstatus. **PRGM** og linie 000 udlæses.

Programlinier. Programhukommelsen er opdelt i *linier*. Et særligt register holder styr på, hvilken programlinie der er aktiv. Dette register kaldes *programpilen* - det peger på den linie, der arbejdes i.

Linie 000 udgør starten (eller toppen) af hukommelsen og kan ikke bruges til at lagre information. Den første egentlige programlinie er linie 001. Når der ikke er lagret noget program, eksisterer der ikke andre linier end 000 – program-pilen er med andre ord nulstillet.

De kan i princippet lade et program starte fra enhver eksisterende programlinie (*nnn*), men det er sikrest at lade et uafhængigt program (i modsætning til et underprogram) starte fra linie 001. Efterhånden som De indtaster programmet, vil de eksisterende programlinier blive skubbet nedad i hukommelsen.

[GTO] [CHS] 000 flytter program-pilen til linie 000, uanset om HP-15C er i normal- eller programmeringsstatus. [GTO]-instruktionen bliver ikke lagret – der er tale om en ren flytning af program-pilen. I *normalstatus* vil [f] CLEAR [PRGM] sætte program-pilen til linie 000, uden at slette program-hukommelsen.

De kan også bruge [f] CLEAR [PRGM] i *programmeringsstatus*, men i så fald bliver programhukommelsen samtidig slettet.

Programstart. En etiket (*label*) – [f] [LBL] efterfulgt af et bogstav ([A] til [E] – *alfaetiket*) eller et tal (0 til 9 eller .0 til .9 – *numerisk etiket*) – bruges til at definere starten på et program eller underprogram. Brugen af etiketter giver Dem en nem mulighed for at vælge mellem flere samtidigt lagrede programmer.

Indtastning

Lyspanel

[f] CLEAR [PRGM]

000 –

Sletter programhukommelsen og sætter program-pilen ved linie 000.

[f] [LBL] [A]

001 – 42.21.11

Det er ikke nødvendigt at trykke på [f] inden [A], når De netop har brugt [f] [LBL].

Lagring af et program. Enhver indtastning* – funktion eller tal – bliver lagret i programhukommelsen som en programinstruktion.

* Undtagen de ikke-programmérbare funktioner, som er anført side 79.

Indtastning**Lyspanel**

2	002 –	2	} Med h i X-registret, vil linierne 002-008 beregne $\sqrt{\frac{2h}{9,8}}$.
$\times$	003 –	20	
9	004 –	9	
.	005 –	48	
8	006 –	8	
$\div$	007 –	10	
$\sqrt{x}$	008 –	11	

Afslutning på program. Et program kan afsluttes på tre forskellige måder:

- $\boxed{g}$ $\boxed{RTN}$ (*return*) afslutter et program, fører programpilen til linie 000 og stopper programudførelsen.
- $\boxed{R/S}$ stopper programkørslen – uden at føre programpilen til linie 000.
- Sidste linie i programhukommelsen er en automatisk $\boxed{RTN}$.

Indtastning**Lyspanel**

$\boxed{g}$ $\boxed{RTN}$	009 – 43 32	Ikke nødvendig hvis sidste instruktion i programhukommelsen.
---------------------------	-------------	--

Stop i selve programmet

$\boxed{f}$ $\boxed{PSE}$ (*pause*) indsætter en pause i programudførelsen, hvor lyspanel-registrets indhold bliver udlæst. (De kan lave længere pauser ved at sætte flere $\boxed{PSE}$ efter hinanden.)

$\boxed{R/S}$ (*run/stop*) stopper programkørslen indtil De starter den igen. Programpilen vil blive stående ved linien med $\boxed{R/S}$. De kan genstarte kørslen ved at trykke på $\boxed{R/S}$ (dvs. i normalstatus).

Kørsel af et program

Normalstatus. Når De er færdig med at indtaste programmet, skal De skifte til normalstatus med $\boxed{g}$ $\boxed{P/R}$. Programudførelse (kørsel) kan kun finde sted i normalstatus.

Indtastning**Lyspanel**

g	P/R
---	-----

Normalstatus; **PRGM** bliver ikke længere vist. (Det udlæste tal er det samme som før skift til programmeringsstatus.)

Programpilens placering ændres ikke af skiftene mellem normal- og programmeringsstatus. Hvis der bliver slukket for HP-15C, vil normalstatus være valgt, når der igen tændes for maskinen.

Udførelse af et program. Tryk

f

alfaetiket eller

GSB

ciffer- eller *alfaetiket* i *normalstatus*. (Bemærk, at en *alfaetiket* er en *bogstavtast* og en *cifferetiket* er en *ciffertast*.) Programpilens hopper hen til den valgte etiket, hvorfra programudførelsen starter. I lyspanelet bliver **running** udlæst blinkende.

Indtastning**Lyspanel**

300.51

300,51

Indtast en værdi for *h*. Lagres som sædvanlig i *X*.

f	A
---	---

7,8313

Resultatet af program "A". (Antal sekunder det tager en genstand at gennemføre et frit fald på 300,51 meter.) Bemærk, at De nu skal bruge

f

 før

A

.

Genstart af program. Brug

R/S

 til at genstarte et program, som blev stoppet af

R/S

.

USER-status. USER-status er en særlig status, der sparer Dem indtastninger ved start af programmer, der indledes med en *alfaetiket*.

f

USER

 bytter om på *f*-funktionerne og primærfunktionerne for

A

 -

E

. De kan derfor starte programmet uden at bruge

f

 eller

GSB

.

Indlæsning af data

Ethvert program skal indeholde instruktioner om hvilke data der skal bruges og hvordan de skal stilles til rådighed for programmet.

Data kan indlæses i normalstatus før programmet køres eller under stop i programmet.

1. **Indtastning før start.** Hvis en variabel skal bruges i programmets første linie, er det bedst at indtaste den, så den står i X ved programstart. Hvis den samme værdi skal bruges senere, kan De lagre den (med `STO`) og senere hente den igen med `RCL` – og begge dele kan gøres i programmet.

Dette er metoden, der blev brugt ovenfor, hvor h blev placeret i X-registret før programmet blev kørt. Det er ikke nødvendigt at bruge `ENTER` fordi programudførelse (her med `f A`) både afslutter talindtastning og aktiverer stakløften. Det ovenstående program ganger derefter indholdet af X-registret (h) med 2.

Stakken giver oven i købet mulighed for at lagre mere end én variabel før start af programmet. Når De betænker, hvorledes stakken bevæger sig under beregninger og hvorledes dens indhold kan manipuleres (med f.eks. `x $\leftrightarrow$ y`), vil De se, at det er muligt at skrive programmer, som udnytter tal, der er lagret i X-, Y-, Z- og T-registre.

2. **Indlæsning under stop.** De kan indlæse data efterhånden som der bliver behov for dem under programkørslen. Sæt en `R/S`-instruktion i programmet, når der er behov for en ny variabel. Når program-pilen møder `R/S`, stopper programkørslen og kontrollen overføres til tastaturet, så De kan indtaste værdien for variabelen. Programmet genstartes med `R/S`.

De skal ikke indtaste variable i selve programmet. Alle ikke-konstante størrelser skal indtastes før eller under kørslen.

Programhukommelsen

Når De første gang tænder for HP-15C efter nulstilling af den kontinuerlige hukommelse, har De rådighed over 322 bytes programhukommelse og 21 dataregistre. De fleste instruktioner fylder én byte, men nogle enkelte optager to. Fordelingen af hukommelsen kan ændres, som forklaret i tillæg C. Den størst mulige programhukommelse er på 448 bytes (med kun de permanente dataregistre – R_0 , R_1 og R_1 – tilbage); det størst mulige antal dataregistre er 67 (hvor der ikke er nogen programhukommelse).

Eksempel

Bedstes Køkken producerer pakker med færdige middagsretter. De enkelte dele leveres i tre dåser af forskellig størrelse: en med spaghetti-sauce, en med reven ost og en med kødboller. Lederen af Bedstes Køkken skal beregne endeflademål, overflade og volumen for de forskellige dåser. Hun skal også bruge det totale endeflademål samt den samlede overflade og volumen for dåserne i en pakke.



Problemet kan løses med et program, som beregner følgende formler:

$$\text{endeflade} = \pi r^2$$

$$\text{volumen} = \text{endeflade} \times \text{højde} = \pi r^2 h$$

$$\text{overflade} = 2 \times \text{endeflade} + \text{krum overflade} = 2\pi r^2 + 2\pi r h$$

Radius, r	Højde, h	Endeflade	Volumen	Overflade
2,5 cm	8,0 cm	?	?	?
4,0	10,5	?	?	?
4,5	4,0	?	?	?
TOTALER		?	?	?

Metode:

1. Indtast en r -værdi og gem den til senere brug. Beregn endefladerne (πr^2) og gem resultatet til senere brug. Læg endelig endefladerne til et register, som bruges til at summere endefladerne.
2. Indtast h og beregn voluminet ($\pi r^2 h$). Læg værdien til et register, som bruges til at summere voluminerne.
3. Fremkald r , divider voluminet med r og gang med 2, så De får beregnet den krumme overflade. Fremkald endefladerne, gang den med 2 og læg resultatet til den krumme overflade. Læg resultatet til et register, der summerer overfladerne.

De skal ikke lagre aktuelle data under programmeringen – kun gøre klar til deres indtastning og anvendelse. Radier og højder vil variere og skal derfor indtastes før/under programkørslen.

Indtast følgende program som løser ovenstående problem. I lyspanelet bliver der vist linienummer og tastkoder for hver enkelt instruktion (se betydningen af tastkoderne under Yderligere information).

Indtastning	Lyspanel	
[g] [P/R]	000 –	Skifter til programmeringsstatus (PRGM bliver vist).
[f] CLEAR [PRGM]	000 –	Sletter programhukommelsen og sætter programpilen ved linie 000.
[f] [LBL] [A]	001 – 42.21.11	Giver programmet navnet "A".
[STO] 0	002 – 44 0	Lagrer indholdet af X i R ₀ , r skal være lagret i X før start af programmet.
[g] [x²]	003 – 43 11	Kvadrerer indholdet af X (dvs. r).
[g] [π]	004 – 43 26	πr^2 . ENDEFLADE for en dåse.
[x]	005 – 20	
[STO] 4	006 – 44 4	Lagrer ENDEFLADE i R ₄ .
[STO] [+] 1	007 – 44.40. 1	Danner sum af alle ENDEFLADE-resultater i R ₁ .
[R/S]	008 – 31	Stopper og udlæser ENDEFLADE. Giver mulighed for indtastning af h.
[x]	009 – 20	Ganger h med ENDEFLADE, så VOLUMEN beregnes.
[f] [PSE]	010 – 42 31	Stopper kort øjeblik og udlæser VOLUMEN.
[STO] [+] 2	011 – 44.40. 2	Danner sum af alle VOLUMEN-resultater i R ₂ .
[RCL] 0	012 – 45 0	Fremkald af r.

Indtastning	Lyspanel	
$\boxed{\div}$	013 – 10	VOLUMEN divideres med r .
2	014 – 2	
$\boxed{\times}$	015 – 20	$2\pi rh$: KRUM OVERFLADE.
$\boxed{\text{RCL}}$ 4	016 – 45 4	Fremkald af ENDEFLADE.
2	017 – 2	} Ganger endeflade med 2.
$\boxed{\times}$	018 – 20	
$\boxed{+}$	019 – 40	KRUM OVERFLADE + ENDEFLADER = OVERFLADE.
$\boxed{\text{STO}}$ $\boxed{+}$ 3	020 – 44.40. 3	Danner sum af alle OVERFLADE-værdier i R_3 .
$\boxed{\text{g}}$ $\boxed{\text{RTN}}$	021 – 43 32	Afslutning på programmet – returhop til linie 000.

Lad os nu prøve at køre programmet.

Indtastning	Lyspanel	
$\boxed{\text{g}}$ $\boxed{\text{P/R}}$		Skifter til normalstatus. (PRGM slukkes.)
$\boxed{\text{f}}$ CLEAR $\boxed{\text{REG}}$		Sletter <i>alle</i> dataregistre. Lyspanelet påvirkes ikke.
2.5	2,5	Indtast r for første dåse.
$\boxed{\text{f}}$ $\boxed{\text{A}}$	19,6350	Start program A.
(eller $\boxed{\text{GSB}}$ $\boxed{\text{A}}$)		ENDEFLADE udlæses efter running .
8	8	Indtast h for første dåse.
$\boxed{\text{R/S}}$	157,0796 164,9336	Genstart program. VOLUMEN af første dåse. OVERFLADE af første dåse.
4	4	Indtast r for anden dåse.
$\boxed{\text{R/S}}$	50,2655	ENDEFLADE af anden dåse.

Indtastning	Lyspanel	
10.5 R/S	10,5 527,7876 364,4247	Indtast <i>h</i> for anden dåse. VOLUMEN af anden dåse. OVERFLADE af anden dåse.
4.5 R/S	4,5 63,6173	Indtast <i>r</i> for tredje dåse. ENDEFLADE af tredje dåse.
4 R/S	4 254,4690 240,3318	Indtast <i>h</i> for tredje dåse. VOLUMEN af tredje dåse. OVERFLADE af tredje dåse.
RCL 1	133,5177	Sum af ENDEFLADER.
RCL 2	939,3362	Sum af VOLUMINER.
RCL 3	769,6902	Sum af OVERFLADER.

Dette program illustrerer de grundlæggende programmeringsprincipper. Det viser desuden, hvorledes data kan manipuleres i programmerings- og normalstatus ved hjælp af indtastning, lagring, fremkald (indlæsning og udlæsning) med ENTER, STO og RCL samt registeraritmetik og programmerede stop.

Yderligere information

Programinstruktioner

Hvert ciffer, decimaltegn og funktion betragtes som en *instruktion* og lagres derfor som en *programlinie* i programhukommelsen. En instruktion kan indeholde en forvalgstast (som f.eks. f, STO, RCL, GTO eller LBL), men vil stadig kun optage en enkelt linie. De fleste instruktioner optager én byte af lagerpladsen, nogle enkelte dog to bytes. I tillæg C kan De se en fuldstændig liste over funktioner, som bruger to bytes ved lagring.

Tastkoder

Hver af tasterne på HP-15C's tastatur (med undtagelse af cifertasterne 0 til 9) identificeres ved hjælp af en to-cifret "tastkode", som svarer til tastens placering på tastaturet (der opfattes som en matrix).

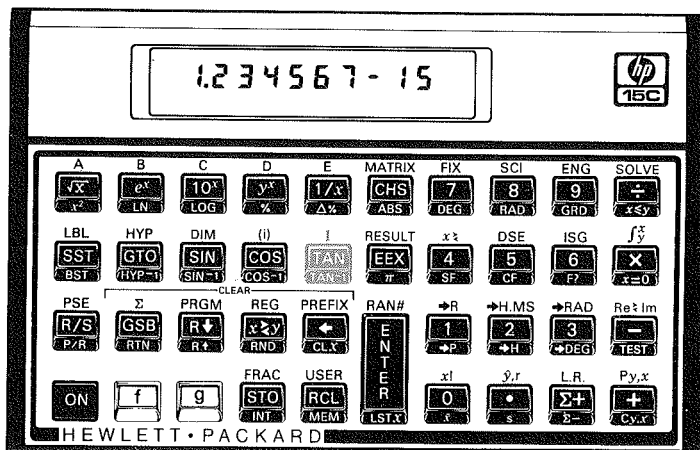
Instruktion

STO + 1
f DSE I

Tastkode

006 - 44.40. 1 6. programlinie
XXX - 42. 5.25 DSE er blot "5".

Det første ciffer angiver rækken (fra 1 til 4, fra top til bund), og det andet ciffer angiver søjlen (1, 2, ..., 9, 0 fra venstre mod højre). Undtagelse: tastkoden for en cifertast er cifret selv.



Tastkode 25: anden række, femte tast

Splitning af lager

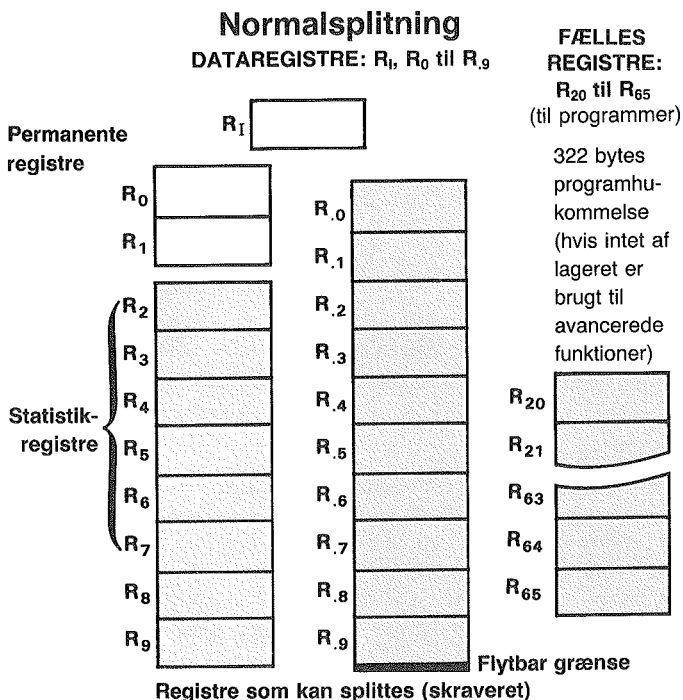
Det er ikke nødvendigt at De forstår mekanismerne bag lagersplitningen, hvis De blot vil bruge HP-15C som en avanceret regnemaskine. Men hvis De ønsker at udnytte programmeringsfaciliteterne, er det nødvendigt, at De har en detaljeret forståelse af hukommelsens opbygning og dynamik. Jo mere De programmerer, desto mere værdifuld vil en sådan indsigt være. Splitning af lager bliver meget grundigt gennemgået i tillæg C, Splitning.

Hvis De nogen sinde får fejlmeddelelsen **Error 10**, betyder det, at De er

stødt på grænsen for HP-15C's lagerkapacitet. Men hvis De har lært selv at ændre fordelingen af den tilgængelige lagerplads, vil De dels have en ringere risiko for at møde begrænsninger, dels kunne udnytte kapaciteten bedre.

HP-15C's lager eller hukommelse består af 67 registre (R_0 til R_{65} samt indeksregistret), som er fordelt mellem registre til lagring af data og registre til lagring af programinstruktioner/instruktioner til brug ved de avancerede matematiske funktioner. Normalsplitningen er:

- 46 registre til programinstruktioner og de avancerede funktioner ($\boxed{\text{SOLVE}}$, $\boxed{\int}$, den imaginære stak og $\boxed{\text{MATRIX}}$). Hvert register svarer til 7 bytes, så De har rådighed over 322 bytes programhukommelse, hvis der ikke er afsat plads til avancerede funktioner.
- 21 registre til lagring af data (R_0 til R_9 , R_{10} til R_{19} og indeksregistret).



Lagerets splitning ændres ved, at De angiver, hvilket dataregister der skal være det sidste (dvs. have den største/højeste adresse); alle øvrige registre vil kunne bruges til programmering og avancerede funktioner.

Indtastning	Lyspanel	
60 [f] [DIM] [(i)] *	60,0000	R ₆₀ og derunder udlagt som dataregistre; fem (R ₆₁ til R ₆₅) til programmering.
1 [f] [DIM] [(i)]	1,0000	R ₀ og R ₁ udlægges som dataregistre; R ₂ til R ₆₅ kan herefter bruges til program(mer) eller til de avancerede funktioner.
19 [f] [DIM] [(i)]	19,0000	Normalsplitning: R ₁₉ (R ₉) og derunder er dataregistre; R ₂₀ til R ₆₅ kan bruges til programmer og avancerede funktioner.**
[RCL] [DIM] [(i)]	19,0000	Udlæser adressen for dataregistret med den største adresse.

[DIM] og [MEM] (*splitning*) bliver nærmere omtalt i tillæg C.

Husk, at der vil blive givet en fejlmeddelelse (*med ovenstående splitning*) i følgende tilfælde:

1. Hvis De prøver at anvende et dataregister med højere adresse end 19 (.9) – den største dataregisteradresse under den givne splitning (**Error 3**).
2. Hvis De har brugt 322 bytes programhukommelse og forsøger at lagre endnu en instruktion (**Error 4**).

* På side 77 forklares, hvornår De kan vælge at undlade at bruge [f] efter brug af en anden forvalgstast – forkortet indtastning.

** Ved ændring af splitning og ved indirekte adressering bliver registre R₀ til R₉ omtalt som R₁₀ til R₁₉.

3. Hvis De forsøger at anvende en af de avancerede funktioner uden at have tilstrækkelig lagerplads (**Error 10**).

Programgrænser

Automatisk slut. Ikke ethvert program behøver at blive afsluttet af **RTN** eller **R/S**. Hvis De står ved slutningen af programhukommelsen (dvs. ved den sidste byte), er der en *automatisk* **RTN**-instruktion, så det ikke er nødvendigt for Dem at lagre en. På denne måde kan De spare en programlinje. På den anden side kan De "afslutte" et program ved simpelt hen at flytte programpilten til et andet programafsnit med **GTO** (se afsnit 7).

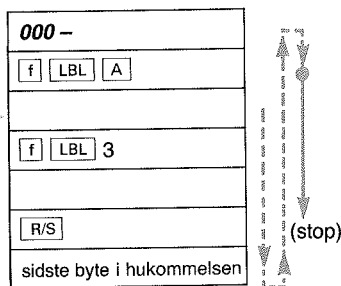
Etiketter. Etiketter i et program eller et underprogram er afmærkninger, som viser, hvor et bestemt programafsnit starter. Efter **f** etiket eller **GSB** etiket vil programpilten lede fremad i programhukommelsen indtil den finder den specificerede etiket.

Hvis etiketten er placeret foran kaldet, vil programpilten springe tilbage til første programlinje og fortsætte søgningen derfra, når den har nået sidste programlinje i hukommelsen. Når den specificerede etiket mødes, stopper søgningen, og programkørslen fortsætter.

Når en etiket mødes under udførelsen af et program, har den ingen effekt; programpilten fortsætter blot videre. De kan derfor frit bruge etiketter til at afmærke programafsnit og underprogrammer midt i selve hovedprogrammet.

Da programpilten kun kan søge i én retning (fremad) er det muligt (om end ikke tilrådeligt) at anvende samme etiket flere gange. Programudførelsen fortsættes altid fra den etiket, der står nærmest efter kaldet.

Hvis et **f** **A**-kald udføres her, vil programpilten starte søgningen herfra og ned gennem hele hukommelsen, hoppe op til linie 000 og stoppe ved etiket "A". Programkørslen genstarter med instruktionen lige efter **f** **LBL** **A**.



Uventede stop

Tryk på vilkårlig tast. Programkørslen stopper hvis De trykker på en af tasterne. Dog vil den operation, der er i gang, blive gennemført før programmet stopper.

Fejlstop. Programkørslen stopper straks, når der bliver forsøgt udført en ugyldig operation (dvs. en operation, som resulterer i en fejlmeddelelse af typen **Error n**).

Fejlmeddelelsen slettes af tryk på en vilkårlig tast. Da programpilen er placeret ud for den instruktion, der gav anledning til fejlstoppet, kan De inspicere programmet ved at skifte til programmeringsstatus.

Hvis tallet i lyspanelet blinker efter et fejlstop, betyder det, at der er indtrådt overløb (side 60). Tryk på ,  eller   9.

Forkortet indtastning

I nogle tilfælde kan De undlade at bruge forvalgstasten . Denne *forkortede indtastning* kan anvendes, når  følger efter en anden forvalgstast. (Side 19 er alle forvalgstasterne opført.)

F.eks. bliver     til   ,     bliver til    og    bliver til  . Det giver ikke anledning til nogen flertydighed, at De fjerner  -  er den eneste logiske mulighed i de givne sammenhænge. Tastkoden for  vil ikke blive registreret, hvis De alligevel bruger tasten.

USER-status

USER-status er en simpel bekvemmelighed, der sparer Dem indtastninger, når De arbejder med programmer, der kaldes (aktiveres) fra tastaturet.   bytter om på de primære og de f-sekundære funktioner for tasterne  til . I USER-status (hvor **USER** er tændt) gælder således:

 sekundær	→	A	B	C	D	E
Primær	→					
 sekundær	→	x ²	LN	LOG	%	Δ%

Næste tryk på   ophæver USER-status.

Polynomier og Horner's metode

I nogle udtryk, som f.eks. polynomier, anvendes samme variabel adskillige gange. F.eks. bliver variabelen x anvendt fire gange i udtrykket:

$$f(x) = Ax^4 + Bx^3 + Cx^2 + Dx + E$$

Et program til direkte beregning af dette udtryk skal hente variabelen fire gange fra et dataregister. Der findes en kortere og generelt mere effektiv metode, hvor rullestakken fyldes med variabelen (se afsnittet om konstant-aritmetik side 41).

Horner's metode er en nyttig metode til omstrukturering af polynomier, således at der spares programplads og beregningstid. Metoden er specielt anvendelig i forbindelse med **SOLVE** og **↵**, som begge er omfattende rutiner med gentagne kald af underprogrammer.

Metoden går ud på, at man omskriver polynomiet med parenteser således, at der ikke optræder eksponenter større end 1:

$$\begin{aligned} & Ax^4 + Bx^3 + Cx^2 + Dx + E \\ & (Ax^3 + Bx^2 + Cx + D)x + E \\ & ((Ax^2 + Bx + C)x + D)x + E \\ & (((Ax + B)x + C)x + D)x + E \end{aligned}$$

Eksempel: Skriv et program, der beregner $5x^4 + 2x^3$ som $((5x + 2)x)x$. Kør programmet med $x = 7$.

Indtastning

g **P/R**

f **LBL** **B**

5

×

2

+

×

×

×

Lyspanel

000 –

001 – 42.21.12

002 – 5

003 – 20

004 – 2

005 – 40

006 – 20

007 – 20

008 – 20

Forudsætter, at program-pilen står ud for linie 000. Hvis ikke, slet da programhukommelsen.

$5x$.

$5x + 2$.

$(5x + 2)x$.

$(5x + 2)x^2$.

$(5x + 2)x^3$.

Indtastning

g RTN

g P/R

7 ENTER ENTER

ENTER

f B

Lyspanel

009 – 43 32

7,0000

12.691,0000

Skifter til normalstatus. Det tidligere indhold af X bliver udlæst.

Lagrer 7 i alle stakregistrerne.

Funktioner, som ikke kan programmeres

Når HP-15C er i programmeringsstatus kan næsten enhver tastfunktion lagres som en programinstruktion. Kun de følgende funktioner kan ikke indgå i et program:

f CLEAR PREFIX

f CLEAR PRGM

f (i)

f USER

g BST

g MEM

g P/R

GTO CHS nnn

SST

←

ON / .

ON / -

Opgaver

1. Et produktionskollektiv har fået installeret en lille sirene, der lyder hver dag kl. 12 middag. 3,2 meter fra sirenen er lydniveauet 138 decibel. Skriv et program, som bestemmer lydniveauet i forskellige afstande fra sirenen.

Brug formelen $L = L_0 - 20 \log(r/r_0)$, hvor:

L_0 er det kendte lydniveau (138 dB) i et punkt nær lydkilden;

r_0 er afstanden fra dette punkt til kilden;

L er lydniveauet i et andet punkt;

r er afstanden fra lydkilden til dette andet punkt (i meter).

Bestem lydniveauet 3 km fra lydkilden ($r = 3.000$ meter).

En mulig løsning er:

g P/R f LBL C 3,2 ÷ g LOG 20 × CHS 138 +

g RTN g P/R, som fylder 15 programtrin og optager 15 bytes programplads. Dette program startes med f C efter indtastning af r .

Opgaven kan løses mere generelt ved at udelade konstanterne i programmet, og i stedet lagre dem i rullestakken inden start. Herved reduceres programmet til 9 linier og 9 bytes. I den modificerede form startes programmet med $\boxed{f} \boxed{C}$ efter $L_0 \boxed{\text{ENTER}} r \boxed{\text{ENTER}} r_0$.

(Svar: $L = 78,5606 \text{ dB}$ for $r = 3.000 \text{ meter}$.)

2. En "typisk" stor tomat vejer 200 gram, hvoraf ca. 188 g (94%) er vand. En tomatdyrker prøver at fremstille tomater med et mindre vandindhold. Skriv et program, som beregner den procentiske normforskel i vandindholdet, dvs. i forhold til vandindholdet i en typisk tomat. Brug et programmeret stop til indtastning af den nye tomat.

Hvor stor er den procentiske normforskel i vandindholdet for en tomat der vejer 230 g, hvoraf 205 g er vand?

En mulig løsning er:

$\boxed{f} \boxed{\text{LBL}} \boxed{D} .94 \boxed{\text{ENTER}} \boxed{\text{R/S}}$ (indtast vægten af vandet i den nye tomat) $\boxed{\text{ENTER}} \boxed{\text{R/S}}$ (indtast totalvægten for den nye tomat) $\boxed{+}$
 $\boxed{g} \boxed{\Delta\%} \boxed{g} \boxed{\text{RTN}}$, som fylder 11 programlinier og optager 11 bytes.

(Svar: for 230 g tomaten er den procentiske ændring i vandindholdet $- 5,1804\%$.)

Editering

Der kan være mange grunde til at man ønsker at ændre et program, som allerede er lagret i programhukommelsen: man kan ønske at fjerne eller indsætte en instruktion (f.eks. **[f]** **[PSE]**) eller man har måske ligefrem begået en fejl... HP-15C har en lang række funktioner og rettefaciliteter, som gør denne *editering* så smertefri som muligt.

Mekanikken

Enhver programændring kræver to trin: De skal føre programpilten hen til den programlinje, hvis indhold skal ændres, og De skal udføre selve ændringen.

Hop til bestemt linie i programhukommelsen

Hopinstruktionen Go To. **[GTO]** **[CHS]** *nnn* fører programpilten hen til linie *nnn*, uanset om maskinen er i normal- eller programmeringsstatus (**PRGM** vises). Denne tastekvens kan *ikke* programmeres, den har udelukkende til formål at give Dem mulighed for manuelt, men hurtigt, at sætte programpilten ved en ønsket linie. Linienummeret skal være et trecifret tal mellem 000 og 448.

Skridtfunktionen. **[SST]** (*single step*) bruges til at føre programpilten én linie frem i programhukommelsen. Funktionen er ikke programmerbar.

I programmeringsstatus: **[SST]** fører programpilten frem til næste programlinje og udlæser linienummer og tastkode. Instruktionen bliver *ikke* udført. Hvis De holder tasten nedtrykket, vil programpilten løbe ned gennem programhukommelsen indtil De slipper tasten.

I normalstatus: Når **[SST]** holdes nedtrykket, udlæses indholdet af den programlinje, som programpilten peger på. Når tasten slippes, bliver instruktionen udført og resultatet udlæst. Herefter går programpilten videre til næste programlinje.

Bakfunktionen. [BST] (*back step*) flytter programpilén én linie tilbage i programhukommelsen, uanset om maskinen er i programmerings- eller normalstatus. Så længe [BST] holdes nedtrykket, vil programpilén løbe baglæns gennem programhukommelsen. Bakfunktionen, som ikke er programmerbar, udfører *ikke* de lagrede instruktioner.

Sletning af programlinier

Programlinier kan slettes med , når maskinen er i *programmeringsstatus*. De skal føre programpilén frem til den linie, der skal slettes, og derefter trykke på . De efterfølgende instruktioner trækkes op i hukommelsen og bliver derved omnummereret.

I normalstatus har  ingen effekt på programhukommelsen, men bruges alene til sletning af cifre og tal i lyspanelregistret (se side 21).

Indsættelse af programlinier

Instruktioner kan indsættes i et eksisterende program ved at De fører programpilén frem til linien umiddelbart før det punkt, hvor den ny instruktion skal lagres. Enhver instruktion, der indtastes, vil blive lagret i programlinien efter den, der udlæses under indtastningen. Det betyder, at hvis De skal *ændre* en instruktion, skal De først slette den oprindelige med  og derefter indtaste den nye.

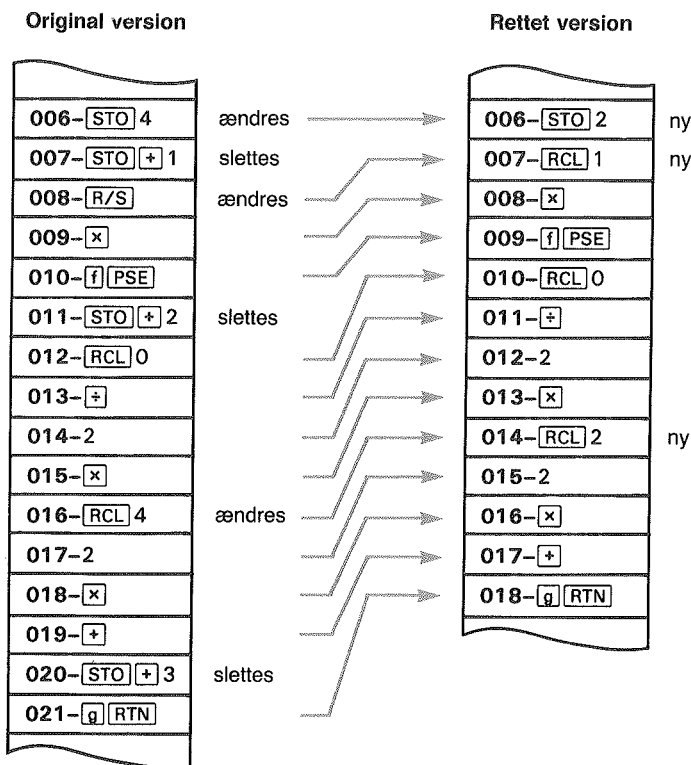
Eksempler

Lad os igen se på volumen-programmet fra side 69 i afsnit 6 og lave et par ændringer. (Det forudsættes at programmet ligger i programhukommelsen fra linie 001.)

Sletning af linier. Hvis det ikke er nødvendigt med endeflade-, volumen- og overfladesummerne, kan vi slette dataregisteroperationerne i linierne 007, 011 og 020.

Ændringer. For at undgå at skulle stoppe programmet, så højden (h) kan indtastes, vil vi ændre [R/S] til [RCL] 1 (R_1 bruges ikke længere, nu hvor vi har opgivet summerne) og skal således lagre h i R_1 før start af programmet. For samtidig at rydde op i brugen af registre vil vi ændre [STO] 4 i linie 006 til [STO] 2 og [RCL] 4 (gammel linie 016) til [RCL] 2 – da vi alligevel ikke længere bruger R_2 og R_3 .

Editeringsprocessen er vist på diagramform på næste side.



Lad os starte fra slutningen af programmet og arbejde os tilbage. På denne måde undgår De at linesletninger påvirker de øvrige linienumre.

Indtastning

[g] **[P/R]**

Brug **[SST]** eller

[GTO] **[CHS]** 020

Lyspanel

000 -

Programmeringsstatus.

020 - 44.40. 3

Flytter programpilen til linie 020 (**[STO]** **[+]** 3).

Indtastning**Lyspanel**

019 – 40

(holdes nede) **016 – 45 4**

015 – 20

2 **016 – 45 2**

Hold eller

011 **011 – 44.40. 2**

010 – 42 31

(holdes nede) **008 – 31**

007 – 44.40. 1

1 **008 – 45 1**

007 – 44.40. 1

006 – 44 4

005 – 20

2 **006 – 44 2**

Linie 020 slettes.

Den næste linie der skal editeres er 016 (4).

Linie 016 slettes.

Linie 016 ændres til 2.

Linie 011 udlæses (2).

Linie 011 slettes.

Baktasten bruges tilbage til linie 008 ().

slettes.

Linie 008 ændres til 1.

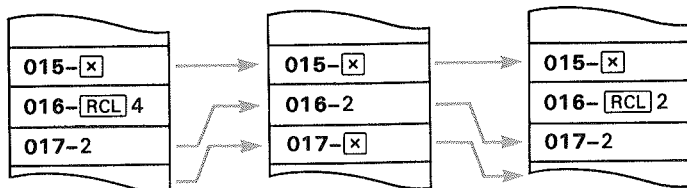
Tilbage til linie 007 med baktasten.

Linie 007 (1) slettes.

Linie 006 (4) slettes.

Ændring til 2.

Udskiftningen af en linie foregår således:



Yderligere information

Linievis udførelse

Linievis udførelse af program. Hvis De ønsker at kontrollere indholdet af et program eller den instruktion, der er lagret i en bestemt programlinje, kan De bruge *skridttasten* til at føre programpilens gennem programhukommelsen – når maskinen er i programmeringsstatus.

Hvis en programkørsel resulterer i fejl eller hvis De har mistanke om, at der er fejl i et programafsnit, kan De kontrollere programmet ved at udføre det linie for linie. Dette gøres med **[SST]** i *normalstatus*.

Indtastning

Lyspanel

[g] **[P/R]**

[f] **CLEAR** **[REG]**

[GTO] **[A]**

8 **[STO]** 1

2.5

[SST] (holdes nede)

(slippes)

[SST]

[SST]

[SST]

[SST]

8,0000

2,5

001 – 42.21.11

2,5000

002 – 44 0

2,5000

003 – 43 11

6,2500

004 – 43 26

3,1416

005 – 20

19,6350

Normalstatus.

Sletter dataregistrene

Flytter programpilen til etiket A.

Lagring af dåsehøjde.

Indtastning af bundradius.

Tastkode i linie 001 (etiket A).

Resultat efter udførelse af linie 001.

[STO] 0.

Resultat.

[g] **[x²]**.

Resultat.

[g] **[π]**

Resultat.

[x]

Resultat: endeflade for dåse.

Tilbageføring. **[SST]** fører aldrig programpilen ind i den del af programhukommelsen, som endnu ikke er taget i brug. I stedet føres programpilen tilbage til linie 000, når pilen når den sidst brugte byte i hukommelsen. (I normalstatus vil **[SST]** dog udføre den sidst lagrede instruktion – som typisk vil være **[g]** **[RTN]**, **[GTO]** eller **[GSB]**.)

Programpilens placering

Husk at programpilens placering ikke ændres af skift mellem normal- og programmeringsstatus og heller ikke af tænd/sluk. Når der skiftes til programmeringsstatus, vil programpilen stå, hvor De efterlod den. (Hvis De har udført et program, som sluttede med **[g]** **[RTN]**, vil programpilen stå ved linie 000.) Det betyder specielt, at hvis den automatiske batterisparer slukker for lyspanelet medens De programmerer, skal De blot tænde for HP-15C og skifte til programmeringsstatus (maskinen er altid i normalstatus, når der tændes for den).

Indsættelse og sletning af programlinier

Efter indsættelse af en ny programlinje vil program-pilen pege på den nye linie. Efter sletning af en programlinje vil program-pilen pege på linien før den, der blev slettet.

Hvis al den tilgængelige plads er opbrugt, vil HP-15C ikke acceptere forsøg på indsættelse af en ny programlinje og reagerer med fejlmeddelelsen **Error 4**.

Initialisering som led i programkørsel

Dataregistrenes indhold og valg af udlæsningsformat osv. vil få indflydelse på kørslen af programmer, som bruger dataregistre og anvender bestemte formater eller udnytter specielle tilstande. I sådanne tilfælde må De sikre Dem, at registre er nulstillede og at der er foretaget de rette statusvalg. De kan enten gøre det manuelt før starten af en programkørsel, eller lægge en initialiseringsrutine ind i selve programmet. Den sidste løsning er langt den sikreste, men den kan kræve for megen plads.

Følgende funktioner bruges typisk til initialisering af registre og tilstande:

[f] CLEAR [Σ], [f] CLEAR [PRGM], [f] CLEAR [REG], [g] [DEG], [g] [RAD], [g] [GRD], [g] [SF] og [g] [CF].

Opgaver

Det hører til god programmeringsskik at lade være med at bruge den samme etiket mere end én gang i et program. (Det skulle ikke være så svært at følge denne skik, eftersom HP-15C giver mulighed for at bruge 25 forskellige etiketter.) For at sikre, at samme etiket ikke bruges flere gange, kan De slette programhukommelsen før lagringen af hvert nyt program.

1. Følgende program bruges til at bestemme, hvor stor en kapital bliver, når den henstår til en fast rente i et antal terminer. Vi bruger den sædvanlige renteformel: $FV = PV(1 + i)^n$, hvor FV er slutværdien, PV er hovedstolen eller startkapitalen, i er rentefoden (som decimal/termin) og n er antallet af terminer. Indtast først PV og derefter n inden De starter programmet. Der er regnet med en årlig rentesats på 7,5% ($i = 0,075$).

Indtastning

[f] [LBL] [.] 1
 [f] [FIX] 2
 1
 [.]
 0
 7
 5
 [x>y]
 [y^x]
 [×]
 [g] [RTN]

Lyspanel

001 - 42.21., 1
 002 - 42. 7. 2
 003 - 1
 004 - 48
 005 - 0
 006 - 7
 007 - 5
 008 - 34
 009 - 14
 010 - 20
 011 - 43 32

} Rente
 (1 + i)ⁿ
 PV(1 + i)ⁿ

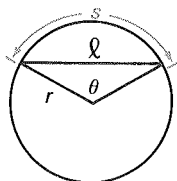
Indtast programmet og find slutværdien af 1.000 kroner som henstår til 7,5% p.a. i 5 år og af 2.300 kr. efter 4 år. De skal bruge [GSB] [.] 1, når De starter programmet. (Svar: 1.435,63 kr. og 3.071,58 kr.)

Indfør de nødvendige ændringer for at bruge rentesatsen 8% p.a.

Brug det editerede program til at finde slutværdien af 500 kr. efter 4 år og af 2.000 kr. efter 10 år. (Svar: 680,24 kr. og 4.317,85 kr.)

2. Skriv et program som beregner længden af korden ℓ over vinklen θ (i grader) i en cirkel med radius r . Brug formelen:

$$\ell = 2r \sin \frac{\theta}{2}.$$



Find ℓ , når $\theta = 30^\circ$ og $r = 25$.

(Svar 12,9410. Et muligt program er: [f] [LBL] [A] [g] [DEG] [f] [FIX] 4 2 [×] [x>y] 2 [÷] [SIN] [×] [g] [RTN]. Programmet forudsætter, at θ er i Y-registret og r i X, når kørslen starter.)

Udfør de ændringer, der er nødvendige for at programmet også kan beregne og udlæse s , længden af cirkelbuen, når θ opgives i radianer. Brug formelen:

$$s = r\theta$$

Udfyld følgende tabel:

θ	r	ℓ	s
45°	50	?	?
90°	100	?	?
270°	100	?	?

(Svar: 38,2683 og 39,2699; 141,4214 og 157,0796; 141,4214 og 471,2389. De kan f.eks. bruge følgende modifikation:

f LBL A g DEG f FIX 4 STO 0 2 × $x \geq y$ STO 1
 2 ÷ SIN × f PSE f PSE RCL 0 RCL 1 f $\rightarrow$ RAD ×
 g RTN .)

Afsnit 8

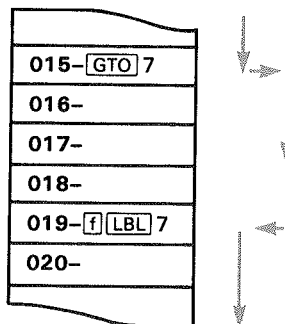
Forgreninger, løkker og flag

Selv om instruktionerne i et program normalt udføres i den lagrede rækkefølge (sekventielt), er det ofte ønskeligt at kunne overføre program-pilen til en linie, som ikke er den næstfølgende. En sådan *forgrening* kan på HP-15C udføres *ubetinget* eller *betinget*, dvs. afhænge af om en bestemt betingelse er opfyldt. Ved at udføre forgrening til en linie, der ligger længere tilbage i programmet, er der mulighed for at arbejde med *løkker*, hvor et programafsnit gentages et vist antal gange eller så længe en bestemt betingelse er opfyldt.

Mekanikken

Forgrening

Hop-instruktionen `GTO`. Simpel forgrening – dvs. ubetinget forgrening – udføres med `GTO` etiket. I et program der kører vil `GTO` flytte program-pilen hen til nærmest efterfølgende linie med den specificerede etiket (`GTO` hopper ikke til et specificeret linienummer).

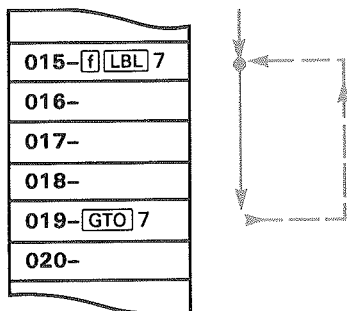


Program-pilen søger fremad i programhukommelsen, går til linie 000, hvis den ikke finder den specificerede etiket i afsnittet fra kaldet til programafslutningen, og fortsætter søgningen fra linie 000.

Løkker. Hvis en `GTO`-instruktion kalder en etiket, som ligger på en linie med et lavere nummer end kaldet (dvs. tidligere i programmet), vil in-

strukturen mellem etiketten og **GTO** blive gentaget – måske i det uendelige...

Denne *løkke* kan styres af en testinstruktion, af **R/S** i selve løkken eller fra tastaturet (De kan altid stoppe programkørslen ved at trykke på en af tasterne).



Sammenligningsoperationer

De kan desuden ændre programforløbet ved hjælp af *sammenligningsoperationer* eller *testinstruktioner*, hvor tallet i X-registret sammenlignes med 0 eller med tallet i Y-registret. Hvis tallene er identiske, siges testen at være opfyldt – svaret på testspørgsmålet er "SAND"; hvis tallene ikke er identiske, er svaret "FALSK". HP-15C giver mulighed for at udføre 12 forskellige sammenligninger. Heraf findes to som direkte tastfunktioner, medens resten aktiveres med **g** **TEST** *n*.*

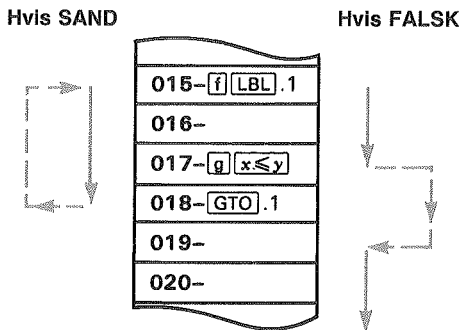
1. Testinstruktioner med særskilt tast: **g** **$x \leq y$** og **g** **$x = 0$** .
2. Testinstruktioner, som aktiveres med **g** **TEST** *n*:

<i>n</i>	Test	<i>n</i>	Test
0	$x \neq 0$	5	$x = y$
1	$x > 0$	6	$x \neq y$
2	$x < 0$	7	$x > y$
3	$x \geq 0$	8	$x < y$
4	$x \leq 0$	9	$x \geq y$

* Fire af disse testinstruktioner kan desuden bruges på komplekse tal – se side 130 i afsnit 11.

Efter en testinstruktion bliver den umiddelbart efterfølgende instruktion udført, hvis resultatet af testen er SAND. Hvis resultatet derimod er FALSK, bliver *næste instruktion ignoreret*. Man vælger ofte at anbringe en `GTO`-instruktion umiddelbart efter testinstruktionen, således at man får *et betinget hop*. Her får man da den specielle regel: HOP hvis SAND.

Programudførelse efter test



Flag

HP-15C har endnu en type testoperationer, som kan bruges under programmering: *flag*. Et flag er en statusindikator, som enten er *sat* (testresultat SAND) eller *slettet* (testresultat FALSK). Også her gælder reglen om, at den efterfølgende instruktion udføres, hvis testen er SAND, og at den springes over, hvis testen er FALSK.

HP-15C har otte *brugerflag*, som er nummererede fra 0 til 7, samt to *systemflag*, flag 8 (kompleksstatus) og flag 9 (overløbsindikation). Systemflagene omtales nærmere i senere afsnit. Alle flag kan sættes, slettes og testes på følgende måde:

- `g SF` *n* sætter flag *n* (0 til 9).
- `g CF` *n* sletter flag *n*.
- `g F?` *n* tester om flag *n* er sat (SAND) eller slettet (FALSK).

Når flag *n* er sat, vil det vedblive at være sat indtil det slettes med enten `g CF` *n* eller den kontinuerlige hukommelse bliver nulstillet.

Eksempler

Eksempel: Forgrening og løkkestruktur

Et radiobiologisk laboratorium ønsker at kunne forudsige radioaktiviteten af en lille mængde ^{131}I , en radioisotop. Skriv et program, som bestemmer aktiviteten for 3-dages intervaller, indtil en given grænse er nået. Formlen for N_t , radioaktiviteten efter t dage, er:

$$N_t = N_0 (2^{-t/k}),$$

hvor $k = 8$ dage, halveringstiden for ^{131}I , og N_0 er startaktiviteten.

Det følgende program bruger en løkke til at beregne den teoretiske strålingsintensitet i millicurie efter 3 dages henfald. Løkken indeholder en testinstruktion, som efter hvert *gennemløb* af løkken kontrollerer, om radioaktiviteten er kommet under en given grænseværdi.

Programmet forudsætter, at t_1 - dagsnummeret for første måledag - er lagret i R_0 , at N_0 - radioaktiviteten i millicurie første dag - er lagret i R_1 , og at grænseværdien for strålingen er lagret i R_2 .



Indtastning

g P/R
f CLEAR PRGM
f LBL A
RCL 0

f PSE
8
÷
CHS
2
x≥y
y^x

Lyspanel

000 -
000 -
001 - 42.21.11
002 - 45 0

003 - 42 31
004 - 8
005 - 10
006 - 16
007 - 2
008 - 34
009 - 14

Programmeringsstatus.
(Kan udelades.)
Startpunkt for løkken.
Fremkald af aktuell t -værdi, som ændres ved hvert gennemløb af løkken.
Pause til udlæsning af t .
 k .
 $-t/k$.
 $2^{-t/k}$.

Indtastning

 RCL $\times$ 1

f PSE

RCL 2

g TEST 9

g RTN

3

STO + 0

GTO A

Lyspanel

010 – 45.20. 1

011 – 42 31

012 – 45 2

013 – 43.30. 9

014 – 43 32

015 – 3

016 – 44.40. 0

017 – 22 11

 Omvendt registeraritmetik med indholdet af R_1 (N_0). Resultatet er N_t , intensiteten i millicurie efter t dage.

 Pause til udl sning af N_t .

Fremkald til X af gr nse-v rdi.

 $x \geq y$? Test, h vor str l ngsin-tensiteten N_t i Y sammenl g-nes med gr nsen i X.

Hvis SAND, er k rslen slut.

Hvis FALSK, forts tter k rslen.

 3 l gges til t i R_0 .

Hop til "A", hvorefter l k-ken genneml bes igen.

Bem rk, at linierne 012 til 014 forhindrer, at programmet forts tter i det uendelige.

Lad os nu k re programmet med $t_1 = 2$ dage, $N_0 = 100$ millicurie og en gr nse p  halvdelen af den oprindelige intensitet: 50 millicurie. (Vi v d, at halveringstiden er 8 dage, s  vi kender allerede resultatet.)

Indtastning

g P/R

2 STO 0

100 STO 1

50 STO 2

f A

Lyspanel

2,0000

100,0000

50,0000

2,0000

84,0896

5,0000

64,8420

8,0000

50,0000

50,0000

Normalstatus.

 t_1 .

 N_0 .

 Gr nse for N_t .

 t_1 .

 N_1 .

 t_2 .

 N_2 .

 t_3 .

 N_3 .

 N_t -gr nse. Programmet stop-per.

Eksempel: Flag

Man kan kategorisere afbetalingslån (annuiteter) på forskellige måder. En måde er at betragte forfaldsdagene for ydelserne. Ved et forudbetalt lån betales ydelserne forud for den kommende beregningstermin (i princippet betales første ydelse samtidig med lånets etablering – en form der ofte ses ved leasing-kontrakter). Ved et efterbetalt lån betales ydelserne efter udløbet af den foregående beregningstermin – en form, der er næsten enerådende ved realkredit og pantebreve med pant i fast ejendom. Vi skal nu skrive et program, hvor et flag bliver brugt til at holde styr på, om en annuitet er for- eller efterbetalt.

En familie har efter en tipsgevinst besluttet at støtte husets datter, som netop står for at starte på en 4-årig uddannelse, med et beløb på 2.500 kroner hver måned. For en sikkerheds skyld skal hele beløbet deponeres på en bankkonto. Hvor stort et beløb skal der indsættes, hvis banken vil give 6% rente p.a., med månedlig rentetilskrivning (0,5% rente/måned).

Formlen er

$$V = -P \left[\frac{1 - (1 + i)^{-n}}{i} \right] (1 + i) \quad \text{hvis pigen trækker pengene i starten af hver måned.}$$

og formelen er

$$V = -P \left[\frac{1 - (1 + i)^{-n}}{i} \right] \quad \text{hvis pigen trækker pengene i slutningen af hver måned.}$$

V er det beløb, som skal deponeres.

P er det beløb, som pigen skal have udbetalt hver måned.

i er rentefoden (som decimal) pr. beregningstermin – i denne sammenhæng den månedlige rentefod.

n er antallet af beregningsterminer.

I det følgende program kan der regnes med begge betalingsformer: både hvor den heldige pige får penge med det samme (forudbetalt) og hvor hun må vente til slutningen af den første måned med at få penge.

Programmet forudsætter at P er lagret i Z , at n er lagret i Y og at i er lagret i X .

Indtastning

g **P/R**
f **LBL** **B**

g **CF** 0

GTO 1
f **LBL** **E**

g **SF** 0

f **LBL** 1

STO 1

1

+

$x \approx y$

CHS

y^x

CHS

1

+

RCL **÷** 1

×

g **F?** 0

g **RTN**

RCL 1

1

+

×

g **RTN**

Lyspanel

000 –

001 – 42.21.12

002 – 43. 5. 0

003 – 22 1

004 – 42.21.15

005 – 43. 4. 0

006 – 42.21. 1

007 – 44 1

008 – 1

009 – 40

010 – 34

011 – 16

012 – 14

013 – 16

014 – 1

015 – 40

016 – 45.10. 1

017 – 20

018 – 43. 6. 0

019 – 43 32

020 – 45 1

021 – 1

022 – 40

023 – 20

024 – 43 32

Programmeringsstatus.

Start ved "B", hvis annuiteten er forudbetalt.

Flag 0 slettes: det er FALSK, at annuiteten er efterbetalt.

Hop til hovedrutine.

Start ved "E", hvis annuiteten er efterbetalt.

Flag 0 sættes: det er SANDT, at annuiteten er efterbetalt.

Start på rutine 1 (hovedrutine).

i lagres fra X .

$(1 + i)$.

Lagrer n i X ; $(1 + i)$ i Y .

$-n$.

$(1 + i)^{-n}$.

$-(1 + i)^{-n}$.

$1 - (1 + i)^{-n}$.

Omvendt registerdivision med $R_1(i)$ for at opnå $[1 - (1 + i)^{-n}]/i$.

Multiplikation med P .

Er flag 0 sat?

Stop, hvis flag 0 er sat (efterbetalt).

Fremkald af i .

$(1 + i)$.

Multiplikation med sidste faktor.

Stop, hvis flag 0 er slettet (forudbetalt).

Kør nu programmet for at finde ud af, hvor stort et beløb der skal ind-sættes på kontoen for gennem 48 måneder at kunne hæve 2.500 kroner hver måned, og banken tilskriver 0,5% rente hver måned. Find først beløbet, hvis pengene udbetales i starten af måneden (forudbetalt) og derefter hvis de udbetales ved månedens udgang (efterbetalt).

Indtastning

[g] [P/R]
2500 [ENTER]

48 [ENTER]
.005

[f] [B]

(Gentag indtastning af data)

[f] [E]

Lyspanel

2.500,0000

48,0000
0,005

106.983,0485

106.450,7945

Skift til normalstatus.

Månedlig udbetaling til pi-gen.

Antal måneder.

Rentesats (som decimaltal).

Beløb, der skal indsættes for at sikre de forudbetalte beta-linger.

Beløb, der skal indsættes for at sikre de efterbetalte beta-linger.

Yderligere information

Go To – ubetinget hop

I modsætning til den ikke-programmérbare sekvens [GTO] [CHS] *nnn* kan den programmérbare sekvens [GTO] *etiket* ikke bruges til at hoppe til et bestemt *linienummer*, men udelukkende til en etiket (en linie som indeholder en instruktion af formen [f] [LBL] *etiket*).* Programkørslen fortsætter fra linien umiddelbart efter etiketten, og vil blive standset af den først forekommende [g] [RTN]. Program-pilen finder således ikke tilbage til det oprindelige programafsnit – medmindre der bruges en [GTO]-instruktion, som fører pilen tilbage til udgangspunktet.

[GTO] *etiket* kan også bruges i normalstatus (dvs. fra tastaturet) til at flytte program-pilen hen til en etiket i programhukommelsen. Denne operation starter ikke programkørsel.

* Det er dog muligt at foretage *indirekte* og ubetinget forgrening til en given linie – se afsnit 10.

Løkker

En løkke er en anvendelse af forgøreling, hvor en $\boxed{\text{GTO}}$ -instruktion bruges til at gentage et afgrænset programafsnit. Løkken kan være uden udgang eller være forsynet med et *betinget udhop*. Desuden kan man indlægge en tæller, som tæller hvor mange gange løkken er udført. Tællerens værdi kan indgå som parameter for det betingede udhop (se eksemplet side 111).

Betinget hop

Betingede hop eller betingede forgørelinger har to hovedanvendelser. Den ene er kontrol af løkker, som omtalt ovenfor. En sammenligningsoperation kan bruges til at teste på en beregnet værdi eller antallet af løkkegennemløb.

Den anden hovedanvendelse er brugen af testinstruktioner til valg mellem alternative programsekvenser. Hvis f.eks. en sælgers provision er progressivt afhængig af salget, vil et provisionsprogram først skulle bestemme, hvilken procentsats der skal anvendes ved provisionsberegningen. Det betyder at salget skal sammenlignes med nogle testværdier. Resultatet af testen afgør da, hvilket programafsnit der skal bruges til provisionsberegningen.

Testinstruktioner. En sammenligningsoperation eller testinstruktion sammenligner tallet i X-registret (x) med enten 0 (f.eks. $\boxed{x=0}$) eller y , dvs. tallet i Y-registret (f.eks. $\boxed{x\leq y}$). For at kunne udføre en $x:y$ -sammenligning skal De derfor have lagret x - og y -værdierne i henholdsvis X og Y. Dette krav kan medføre, at De skal bruge et dataregister til lagring af testværdien (typisk x), som kan hentes frem når selve testen udføres og den værdi, der skal sammenlignes med testværdien, allerede er genereret (dvs. står i X og bliver skubbet op i Y, når testværdien hentes). I nogle tilfælde kan testværdien blive stående i stakken, således at den hentes frem med de sædvanlige funktioner til stakmanipulation ($\boxed{x\geq y}$, $\boxed{R\downarrow}$ og $\boxed{R\uparrow}$).

Sammenligningsoperationer med komplekse tal og matrixdeskriptorer.

Fire af testinstruktionerne kan bruges på komplekse tal og matrixdeskriptorer – det er de fire operationer som tester på identitet: $\boxed{g} \boxed{x=0}$, $\boxed{g} \boxed{\text{TEST}} 0 (x \neq 0)$, $\boxed{g} \boxed{\text{TEST}} 5 (x = y)$ og $\boxed{g} \boxed{\text{TEST}} 6 (x \neq y)$. Se afsnittene 11 og 12 for nærmere oplysninger.

Flag

På samme måde som en sammenligningsoperation kan bruges til at lade programmet træffe beslutninger på grundlag af en sammenligning af to tals indbyrdes størrelse, kan flag bruges til at definere tilstande, både før kørslen og dynamisk under programkørslen.

Flag bliver ofte brugt således, at valget af programstart afgør om et flag skal sættes eller slettes (se eksemplet side 94).

På denne måde er programmet umiddelbart tilpasset to forskellige indatatilstande – f.eks. kan man arbejde med to vinkelmål og bruge flag til at sikre, at beregningerne følger det valgte mål. De kan lade et flag være sat, hvis en omregning skal udføres, og tilsvarende slette flaget, hvis omregningen ikke skal udføres.

Antag at De arbejder med et udtryk, hvor der indgår en temperatur målt i Kelvin, men at De somme tider får temperaturen opgivet i Celsius. De kan bruge et flag til at tillade inddata i såvel Kelvin som Celsius. Programafsnittet kunne f.eks. være:

f **LBL** **C**

Start program med "C" for Celsius.

g **CF** 7

Flag 7 slettes (FALSK).

GTO 1

f **LBL** **D**

Start program med "D" for Kelvin.

g **SF** 7

Flag 7 sættes (SAND).

f **LBL** 1

(Temperaturen forudsættes i X.)

g **F?** 7

Test af flag 7.

GTO 2

Hop til etiket 2, hvis test = SAND.

2

273 lægges til værdien i X, hvis

7

test = FALSK ($K = ^\circ C + 273$).

3

+

f **LBL** 2

Beregningen fortsætter for begge tilfælde.

·
·
·

Systemflagene: Flag 8 og 9

Flag 8. Når flag 8 er sat, er HP-15C i kompleksstatus (se afsnit 11) og C tændt i lyspanelet. Hvis kompleksstatus aktiveres på anden måde, bliver flag 8 automatisk sat. Kompleksstatus kan kun ophæves ved at slette flag 8. Flag 8 slettes på samme måde som andre flag.

Flag 9. Flag 9 bliver automatisk sat, når der indtræder overløb (se side 60). Flag 9 får lyspanelet til at blinke. Hvis et program er under udførelse, vil lyspanelet begynde at blinke, når udførelsen er afsluttet.

Flag 9 kan slettes på tre måder:

- Brug 9 CF 9 (den sædvanlige flagsletningsmetode).
- Tryk på ◀. I denne sammenhæng vil slettetasten udelukkende slette flag 9 og stoppe den blinkende udlæsning.
- Sluk for HP-15C med ON. (Tallet i lyspanelet vil stadig blinke, når De tænder for maskinen efter sluk med den automatiske batterisparer.)

De kan sætte flag 9 manuelt (9 SF 9) så tallet i lyspanelet blinker helt uafhængigt af maskinens overløbsstatus. Det kan gøres under programkontrol, således at særlig vigtige resultater kan blive udlæst blinkende.

Underprogrammer

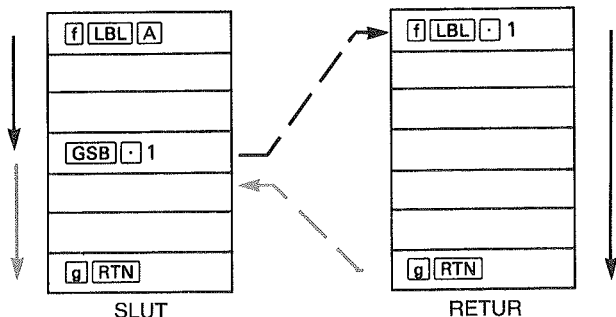
Når en række instruktioner skal bruges flere gange under samme programudførelse, kan der spares programplads ved at lagre instruktionerne i et særligt markeret programafsnit – et *underprogram*.

Mekanikken

Kald af underprogram og returhop

GSB (*go to subroutine*) udføres på samme måde som **GTO** og virker på næsten samme måde. Der er dog én vigtig undtagelse: den lagrer en *returadresse*. **GSB** etiket vil, ligesom **GTO** etiket*, flytte program-pilen hen til den først forekommende linie med den specificerede etiket (**A** til **E**, 0 til 9 eller .0 til .9). Herfra fortsættes programudførelsen indtil pilen møder en **RTN**-instruktion. Herfra springer pilen tilbage til instruktionen umiddelbart efter den sidst aktiverede **GSB**-instruktion, hvorfra programkørslen fortsættes. Dette hop kaldes et *returhop*.

Udførelse af underprogram



Program-pilen føres tilbage til linie 000 og stopper.

Program-pilen føres tilbage til instruktionen umiddelbart *efter* **GSB** . 1.

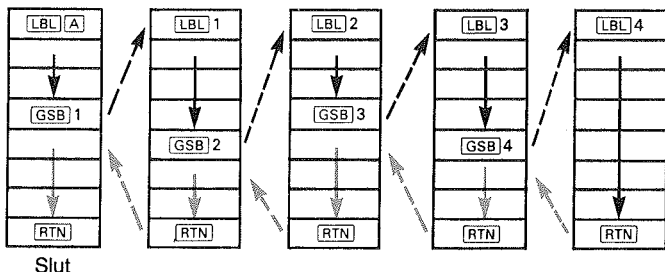
* En **GTO**- eller **GSB**-instruktion, som efterfølges af en alfa-etiket, kan indtastes uden at bruge **f** foran etiket-navnet (se side 77).

Kald af underprogram fra underprogram

Et underprogram kan kalde et andet underprogram, som igen kan kalde et nyt underprogram – og hver gang lagres *returadressen* i et internt register, som har plads til syv adresser.

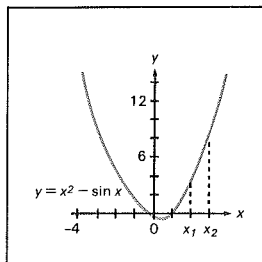
Mekanikken bag underprogramkald i flere niveauer fremgår af nedenstående diagram:

Hovedprogram



Eksempler

Eksempel: Skriv et program, som beregner hældningen af sekanten mellem punkter (x_1, y_1) og (x_2, y_2) på den viste graf, hvor $y = x^2 - \sin x$ (x i radian). Sekantens hældningskoefficient er givet ved:



$$\frac{y_2 - y_1}{x_2 - x_1}, \text{ eller } \frac{(x_2^2 - \sin x_2) - (x_1^2 - \sin x_1)}{x_2 - x_1}$$

Løsningen kræver, at udtrykket for y skal beregnes to gange, én gang for y_1 og én gang for y_2 . Da der er tale om eksakt samme beregning, blot med forskellige argumenter, er der mening i at bruge et underprogram til beregning af y_i .

I det følgende program forudsættes det, at x_1 er lagret i Y og at x_2 er lagret i X.

HOVEDPROGRAM

g P/R

f CLEAR PRGM

(Ikke programmerbar.)

000 -

001 - f LBL 9

Start på hovedprogram.

002 - g RAD

Radian som vinkel mål.

003 - STO 0

Lagring af x_2 i R_0 .004 - $x \geq y$ Lagrer x_1 i X; x_2 i Y.

005 - STO - 0

 $(x_2 - x_1)$ i R_0 .

006 - GSB .3

Hop til underprogram ".3" med x_1 i X.

Retur fra underprogram ".3".

007 - CHS

 $-y_1$.008 - $x \geq y$ Lagring af x_2 i X.

009 - GSB .3

Hop til underprogram ".3" med x_2 i X.

Retur fra underprogram ".3".

010 - +

 $y_2 - y_1$.

011 - RCL ÷ 0

Fremkalder $(x_2 - x_1)$ fra R_0 og
beregner $(y_2 - y_1)/(x_2 - x_1)$.

012 - g RTN

Programafslutning (retur til linie 000).

UNDERPROGRAM

013 - f LBL .3

Start på underprogram ".3".

014 - g x^2 x^2 .

015 - g LSTx

Fremkald x .

016 - SIN

 $\sin x$.

017 - -

 $x^2 - \sin x$; dvs. y .

018 - g RTN

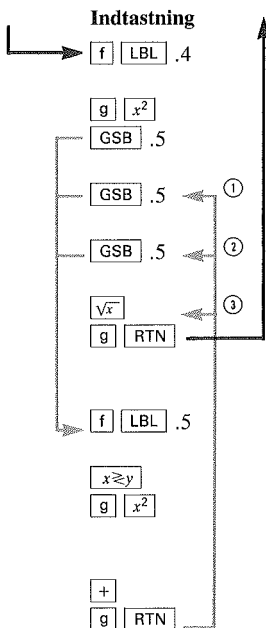
Returhop til hovedprogrammet.

Beregn hældningskoefficienten for følgende værdier af x_1 og x_2 : 0,52 og 1,25; -1 og 1; 0,81 og 0,98. Husk at bruge GSB 9 (og ikke f 9), når De adresserer den numeriske etiket.

(Svar: 1,1507; -0,8415; 1,1652.)

Eksempel: Flere niveauer. I følgende underprogram ".4" bliver udtrykket $\sqrt{x^2 + y^2 + z^2 + t^2}$ beregnet som led i en større beregning i et større program. Underprogrammet kalder et *andet* underprogram (".5"), som udfører kvadreringen.

Programmet udføres efter at variablerne t, z, y og x er lagret i rullestakken (i denne rækkefølge).



Start på 1. underprogram.

x^2 .

Beregner y^2 og

$x^2 + y^2$.

Beregner z^2 og

$x^2 + y^2 + z^2$.

Beregner t^2 og

$x^2 + y^2 + z^2 + t^2$.

$\sqrt{x^2 + y^2 + z^2 + t^2}$.

Slut på underprogram;

programpilens går tilbage til hovedprogram.

Start på underprogram i 2. niveau.

Kvadrerer tallet og

lægger resultatet til den eksisterende sum.

Slut på underprogrammet i

2. niveau; programpilens går tilbage til underprogrammet i 1. niveau.

Hvis De kører underprogrammet (med dets underprogram) med $x = 4,3$, $y = 7,9$, $z = 1,3$ og $t = 8,0$, vil De få 12,1074 som svar efter aktivisering med [GSB] .4.

Yderligere information

Returhoppet

Når program-pilen møder en `RTN` efter `GSB` vil den *lagrede returadresse* sende program-pilen tilbage til linien efter `GSB` og ikke til linie 000. Det er denne mekanisme, der gør underprogrammer så nyttige. De kan selv definere funktioner, som kan bruges flere gange i samme program – uden at de optager meget mere plads end den, der direkte bruges til at definere funktionen.

Den eneste forskel mellem `GSB` og `GTO` er, at `GSB` lagrer en returadresse i den særlige returadressestak. (Man kan egentlig godt sige, at `GTO` blot er et specialtilfælde, som lagrer besked om hop til linie 000.)

Underprogrammer i flere niveauer

Hvis De forsøger at arbejde med underprogrammer i flere end 7 niveauer, vil programkørslen stoppe og fejlmeddelelsen **Error 5** bliver udlæst, når den ottende `GSB` mødes.

Bemærk, at der ikke er nogen som helst begrænsninger på, hvor mange underprogramkald der kan udføres – De må blot ikke have flere end 7 aktive kald ad gangen.

Indeksregistret og løkker

Indeksregistret (R_1) giver betydelige lettelser i forbindelse med avanceret programmering af HP-15C. Udover at kunne bruges til lagring af konstanter og mellemresultater kan indeksregistret bruges til:

- Tælling i og kontrol af løkker.
- Indirekte adressering af dataregistre (også registre ud over R_9 (R_{19})).
- Indirekte adressering af *linienumre* og etiketter.
- Indirekte kontrol af udlæsningsformatet.
- Indirekte kontrol af flag.

I og **(i)**

Direkte kontra indirekte datalagring med indeksregistret

Indeksregistret er et dataregister, som kan bruges direkte med **I** og indirekte med **(i)**.^{*} Forskellen mellem disse to adresseringsformer er vigtig:

I bruger det lagrede tal.

(i) bruger den numeriske værdi af heltalsdelen af det lagrede tal som adresse for et andet dataregister. Denne proces omtales som *indirekte adressering*.

^{*} Bemærk at også de komplekse funktioner og matrixfunktionerne anvender **I** og **(i)** men til andre formål. Se afsnit 11 og 12.

Indirekte programkontrol ved hjælp af indeksregistret

I-tasten bruges til alle former for indirekte adressering, *med undtagelse af* indirekte registeradressering. Det er således **I** (og ikke **(i)**) der bruges til indirekte forgrening, indirekte kontrol af udlæsningsformat og indirekte kontrol af flag.

Løkker

Løkketælling og -kontrol kan udføres af et *vilkårligt dataregister*: fra R_0 til R_9 og indeksregistret. Men løkkekontrol kan desuden udføres indirekte med **(i)**.

Mekanikken

Både **I** og **(i)** kan bruges med forkortet indtastning, hvor **f** udelades (se side 77).

Lagring og fremkald med indeksregistret

Direkte. **STO** **I** og **RCL** **I**. Lagring og fremkald mellem X-registret og indeksregistret udføres på præcis samme måde som for ethvert andet dataregister (side 42).

Indirekte. **STO** (eller **RCL**) **(i)** lagrer i (eller fremkalder fra) det dataregister, hvis adresse er lig den numeriske værdi af heltalsværdien af tallet i indeksregistret (0 til 65). Se tabellen nedenfor.

Indirekte adressering

Hvis R_i indeholder	vil (i) adressere	vil GTO I og GSB I forgrene til*
± 0	R_0	f LBL 0
$\vdots$	$\vdots$	$\vdots$
9	R_9	f LBL 9
10	R_{10}	" " .0
11	R_{11}	" " .1
$\vdots$	$\vdots$	$\vdots$
19	R_{19}	f LBL .9
20	R_{20}	" " A

* Gælder kun for $R_i \geq 0$.

(Fortsættes på næste side.)

Indirekte adressering

Hvis R_i indeholder	vil (i) adressere	vil $\boxed{\text{GTO}} \boxed{i}$ og $\boxed{\text{GSB}} \boxed{i}$ forgrene til*
21	R_{21}	$\boxed{f} \boxed{\text{LBL}} \boxed{B}$
22	R_{22}	" " $\boxed{C}$
23	R_{23}	" " $\boxed{D}$
24	R_{24}	" " $\boxed{E}$
$\vdots$	$\vdots$	—
65	R_{65}	—

* Gælder kun for $R_i \geq 0$.

Aritmetik med indeksregistret

Direkte. $\boxed{\text{STO}}$ eller $\boxed{\text{RCL}} \{ \boxed{+}, \boxed{-}, \boxed{\times}, \boxed{\div} \} \boxed{i}$. Direkte og omvendt registeraritmetik kan udføres med indeksregistret som med ethvert andet dataregister (side 43).

Indirekte. $\boxed{\text{STO}}$ eller $\boxed{\text{RCL}} \{ \boxed{+}, \boxed{-}, \boxed{\times}, \boxed{\div} \} \boxed{(i)}$. Direkte og omvendt registeraritmetik kan udføres med det dataregister hvis adresse er lig den numeriske værdi af heltalsværdien af tallet i indeksregistret (0 til 65). Se tabellen ovenfor.

Ombytning med X-registret

Direkte. $\boxed{f} \boxed{x \geq} \boxed{i}$ ombytter tallene i X-registret og indeksregistret. (Virker på samme måde som $\boxed{x \geq} n$ med registrene 0 til 9.)

Indirekte. $\boxed{f} \boxed{x \geq} \boxed{(i)}$ ombytter tallene i X-registret og det dataregister hvis adresse er lig den numeriske værdi af heltalsværdien af tallet i indeksregistret (0 til 65). Se tabellen ovenfor.

Indirekte forgrening med $\boxed{i}$

$\boxed{i}$ – men ikke $\boxed{(i)}$ – kan bruges til *indirekte* forgrening ($\boxed{\text{GTO}} \boxed{i}$) og kald af underprogrammer ($\boxed{\text{GSB}} \boxed{i}$). (Det er kun den numeriske værdi af heltalsværdien af tallet i R_i , der bliver brugt.) ($\boxed{(i)}$ bliver udelukkende brugt til indirekte adressering af *dataregistre*.)

Til etiket. Hvis tallet i R_1 er *positivt*, vil $\boxed{\text{GTO}} \boxed{\text{I}}$ og $\boxed{\text{GSB}} \boxed{\text{I}}$ føre program-pilen hen til den *etiket*, der numerisk svarer til tallet (se tabel side 106-107).

Hvis f.eks. indeksregistret indeholder 20,00500, vil $\boxed{\text{GTO}} \boxed{\text{I}}$ føre program-pilen til linien med instruktionen $\boxed{\text{f}} \boxed{\text{LBL}} \boxed{\text{A}}$. (Se tabellen side 106.)

Til linienumre. Hvis tallet i R_1 er *negativt*, vil $\boxed{\text{GTO}} \boxed{\text{I}}$ og $\boxed{\text{GSB}} \boxed{\text{I}}$ føre program-pilen til den linie, hvis nummer er lig den numeriske værdi af heltalsværdien af tallet.

Hvis f.eks. R_1 indeholder -20,00500, vil $\boxed{\text{GTO}} \boxed{\text{I}}$ føre program-pilen til linie 020.

Indirekte kontrol af flag med $\boxed{\text{I}}$

$\boxed{\text{SF}} \boxed{\text{I}}$, $\boxed{\text{CF}} \boxed{\text{I}}$ eller $\boxed{\text{F?}} \boxed{\text{I}}$ vil sætte, slette eller teste flaget hvis adresse (0 til 9) er lagret i R_1 (den numeriske værdi af heltalsdelen).

Indirekte kontrol af udlæsningsformatet med $\boxed{\text{I}}$

$\boxed{\text{f}} \boxed{\text{FIX}} \boxed{\text{I}}$, $\boxed{\text{f}} \boxed{\text{SCI}} \boxed{\text{I}}$ og $\boxed{\text{f}} \boxed{\text{ENG}} \boxed{\text{I}}$ vil formattere lyspanelet på sædvanlig vis og bruger den numeriske værdi af heltalsdelen i R_1 til at bestemme antallet af decimaler (se side 57-58).

Løkker med tællere: $\boxed{\text{ISG}}$ og $\boxed{\text{DSE}}$

$\boxed{\text{ISG}}$ (*increment and skip if greater than*) og $\boxed{\text{DSE}}$ (*decrement and skip if less than or equal*) bruges til at styre løkker. $\boxed{\text{ISG}}$ – *inkrementering og skip hvis større end* – tæller antallet af gennemløb ved at lægge et *inkrement* til en *løkketæller* og skippe (overspringe) den efterfølgende linie, hvis tælleren herefter er større end en given *testværdi*. $\boxed{\text{DSE}}$ – *dekrementering og skip hvis lig med eller mindre end* – trækker et *inkrement* fra (*dekrementerer*) løkketælleren og skipper den efterfølgende linie, hvis tælleren herefter er lig med eller mindre end en given *testværdi*.

Både løkketæller og testværdi indgår i et *kontroltal*, som er lagret i et dataregister.

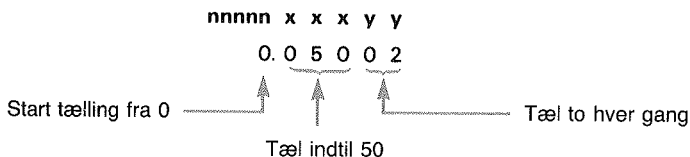
Operationerne har formen $\boxed{\text{f}} \{ \boxed{\text{ISG}}, \boxed{\text{DSE}} \}$ *registeradresse*. Adressen kan være en sædvanlig adresse – dvs. 0 til 9, .0 til .9, $\boxed{\text{I}}$ eller $\boxed{\text{(i)}}$.*

* Undtagen i forbindelse med $\boxed{\text{f}} \boxed{\text{f}}$ (afsnit 14).

Kontroltal. Kontroltallet har følgende format:

	$\pm nnnnn$	den aktuelle tællerværdi,
$nnnnn.xxyy$, hvor	xxx	er test(slut)værdien og
	yy	er inkrementerings/dekrementeringsværdien (skridtværdien).

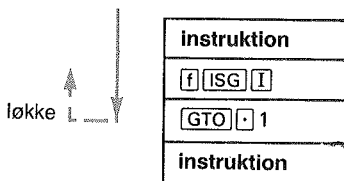
F.eks. vil tallet 0,05002 repræsentere:



[ISG] og [DSE]. Hver gang programpilen møder [ISG] eller [DSE], vil $nnnnn$ (kontroltallets heltalsdel) i det givne dataregister blive inkrementeret eller dekrementeret, således at der på denne måde holdes tal på antallet af gennemløb. Derefter sammenlignes $nnnnn$ med testværdien xxx . Næste programlinje bliver sprunget over, hvis løkketælleren er større end ([ISG]) eller lig med/mindre end ([DSE]) testværdien (xxx). Skridtværdien yy bruges til at angive hvor stort et tal der skal lægges til ([ISG]) eller trækkes fra ([DSE]).

Da der normalt bruges en hopordre efter [ISG] eller [DSE], følger disse funktioner ikke "HOP hvis SAND"-reglen, således som det ellers normalt gælder for sammenligningsoperationer. Derimod gælder "SKIP hvis SAND".

FALSK ($nnnnn \leq xxx$)

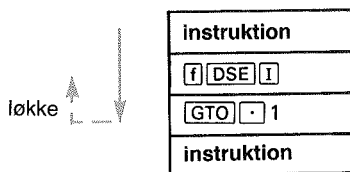


SAND ($nnnnn > xxx$)



Mekanismen bag ISG: Med **nnnnn.xxyy** som kontroltal inkrementeres **nnnnn** til **nnnnn + yy**, som sammenlignes med **xxx**. Hvis **nnnnn > xxx** skippes den efterfølgende programlinie. ISG giver således mulighed for at forlade løkken hvis og kun hvis **nnnnn** bliver større end **xxx**.

FALSK ($nnnnn > xxx$)



SAND ($nnnnn \leq xxx$)



Mekanismen bag DSE: Med **nnnnn.xxyy** som kontroltal dekrementeres **nnnnn** til **nnnnn - yy**, som sammenlignes med **xxx**. Hvis **nnnnn ≤ xxx** skippes den efterfølgende programlinie. DSE giver således mulighed for at forlade løkken hvis og kun hvis **nnnnn** bliver mindre end eller lig med **xxx**.

Nedenstående tabel illustrerer inkrementering og dekrementering af forskellige tællerværdier.

Antal gennemløb

Operation	0	1	2	3	4
ISG	0,00602	2,00602	4,00602	6,00602	8,00602 (skip næste linie)
DSE	6,00002	4,00002	2,00002	0,00002 (skip næste linie)	

Eksempler

Eksempler: Registeroperationer

Lagring og fremkald

Indtastning

f CLEAR REG

12.3456

STO I

Lyspanel

12,3456

12,3456

Nulstiller alle dataregistre.

Lagres i R₁.

Indtastning

7 $\sqrt{x}$
 STO (i)

Lyspanel**2,6458****2,6458**

RCL I
 RCL (i)
 f $x \geq$.2

12,3456**2,6458****2,6458**

Lagring i R_2 ved hjælp af indirekte adressering ($R_1 = 12,3456$).

Fremkald af tallet i R_1 .

Indirekte kald af tallet i R_2 .

Kontrol: samme tal udlæses efter direkte kald af R_2 .

Ombytning med X-registret**Indtastning**

f $x \geq$ I

Lyspanel**12,3456**

RCL I
 f $x \geq$ (i)

2,6458**0,0000**

RCL (i)
 f $x \geq$ 2

2,6458**2,6458**

Ombytning af tallene i R_1 og X.

Nuværende indhold af R_1 .

Ombytning af tallene i R_2 og X.

Kontrol: samme tal udlæses efter direkte kald af R_2 .

Registeraritmetik**Indtastning**

10 STO + I
 RCL I
 g π STO $\div$ (i)

Lyspanel**10,0000****12,6458****3,1416**

RCL (i)
 f $x \geq$.2

0,8422**0,8422**

10 lægges til R_1 .

Nyt indhold af R_1 .

Indholdet af R_2 divideres med π .

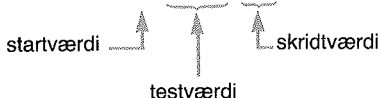
Nyt indhold af R_2 .

Kontrol: direkte kald af R_2 .

Eksempel: Løkkekontrol med DSE

I afsnit 8 (side 92 ff) lavede vi et program, som brugte en løkke til beregning af radioaktivitet. Programmet brugte en test ($x \geq y?$) til at hoppe ud af løkken, når resultatet nåede en vis grænse. Som vi har set i dette afsnit kan man kontrollere løkker på en anden måde: ved hjælp af en løkketæller, som styres af ISG eller DSE.

Nu følger en revision af det oprindelige henfaldsprogram. Denne gang vil vi begrænse programkørslen til tre gennemløb fremfor at sætte en bestemt grænse. Vi vil bruge **DSE** og R_2 med kontroltal 3, 0 0 0 0 1.



Udfør følgende ændringer (med hele det oprindelige program i programhukommelsen): en løkketæller lagres i R_2 og et linienummer lagres i indeksregistret.

Indtastning	Lyspanel	
g P/R	000 –	Programmeringsstatus.
GTO CHS 013	013 – 43.30. 9	Programpilen føres frem til den anden af de to testlinier.
← ←	011 – 42 31	Sletning af linierne 013 og 012.
f DSE 2	012 – 42. 5. 2	Indfør løkketællerfunktionen (kontroltal lagres i R_2 .)
GTO I	013 – 22 25	Hop til givent linienummer (015).

Når løkketælleren i R_2 bliver lig med 0 vil programpilen skippe linie 013 og fortsætte fra linie 014, hvor der er lagret en **RTN**-instruktion, som afslutter programudførelsen. Så længe løkketælleren er større end 0 vil programkørslen fortsætte fra linie 013, hvorfra der forgrenes til linie 015.

Før De starter programmet skal t_1 (dag 1) lagres i R_0 , N_0 (startintensitet) i R_1 , løkketælleren (kontroltal) i R_2 og linienummeret for forgreningen i indeksregistret.

Indtastning	Lyspanel	
g P/R		Normalstatus.
2 STO 0	2,0000	t_1 .
100 STO 1	100,0000	N_0 .
3.00001 STO 2	3,0000	Løkketæller. (Denne instruktion kunne have været programmeret).

Indtastning

15 CHS STO I
f A

Lyspanel

– 15,0000
2,0000

84,0896

5,0000

64,8420

8,0000

50,0000

50,0000

Forgreningslinie.

Programmet kører: løkketæller = 3.

Løkketæller = 2.

Løkketæller = 1.

Løkketæller = 0; programmet er færdigt.

Eksempel: Kontrol af udlæsningsformatet

Følgende program indeholder en pauseinstruktion, hvor et tal udlæses med forskellige fastkommaformater, som ændres ved hjælp af en DSE-løkke.

Indtastning

g P/R
f CLEAR PRGM
f LBL B

9

nnnnn = 9. xxx = 0 og yy = 01 (yy kan ikke være 0).

STO I
f LBL 0
f FIX I
RCL I
f PSE
f DSE I

GTO 0
g TEST 1
GTO 0

g RTN

Udlæser aktuel værdi af nnnnn.

Tallet i R₁ dekrementeres og testes. Skip næste linie, hvis nnnnn ≤ testværdi.

Fortsæt i løkken, hvis nnnnn > testværdi (0).

Tester om tallet i lyspanelet er større end 0.

Løkken fortsætter når nnnnn er blevet 0, men der stadig står 1,0 i lyspanelet.

Udlæsning med alle fastkommaformater:

Indtastning

g P/R

f B

Lyspanel**9,000000000****8,00000000****7,0000000****6,000000****5,00000****4,0000****3,000****2,00****1,0****0,****0.**

Normalstatus

Udlæsning med **f** **PSE**.

Udlæst tal, når programmet stopper.

Yderligere information

Indhold af indeksregistret

Ethvert tal, der lagres i indeksregistret, kan bruges på tre forskellige måder:

- R_1 kan bruges som ethvert andet dataregister – tallet kan manipuleres med de sædvanlige funktioner: kaldes frem, ombyttes med lyspanelet, indgå i aritmetiske operationer osv.
- Tallet kan bruges som kontroltal. Den numeriske værdi af heltalsdelen af tallet i R_1 betragtes som et tal, der er helt uafhængigt af decimaldelen. Det er kun den numeriske værdi af heltalsdelen, der bliver brugt til indirekte forgrening, flagkontrol og formattering. Også decimaldelen bliver brugt ved løkkekontrol, men stadig bruges tallene på hver side af decimaltegnet til helt forskellige formål.*
- **(i)** kan bruges som reference til et andet dataregister. **(i)** anvender det indirekte adresseringssystem, som er beskrevet i tabellerne side 106 og 107. (Det betyder f.eks., at indholdet af det indirekte adresserede register kan bruges som kontroltal.)

* Dette gælder for *ethvert* dataregister, som bruges til indirekte løkkekontrol.

[ISG] og [DSE]

Til løkkekontrol kan heltalsværdien af kontroltallet (løkketælleren) være på indtil 5 cifre (nnnnn.xxxyy). Tælleren (nnnnn) sættes til 0, hvis intet andet specificeres.

Test- eller slutværdien xxx i kontroltallets decimaldel skal altid opgives som et 3-cifret tal, også selv om der er færre betydende cifre. (F.eks. skal "5" opgives som "005".) xxx sættes til 0, hvis ikke andet specificeres. Når [ISG] eller [DSE] mødes, bliver nnnnn internt sammenlignet med xxx.

Skridtværdien yy i kontroltallets decimaldel skal opgives som et 2-cifret tal. yy kan ikke være 0, så hvis der ikke opgives nogen værdi, eller værdien sættes til 00, bliver 01 brugt. Værdien af nnnnn ændres med yy hver gang [ISG] eller [DSE] mødes. Såvel yy som xxx er referenceværdier, som ikke ændres under løkkeudførelsen.

Indirekte kontrol af udlæsningsformatet

Om end De kan bruge indeksregistret til manuel formattering (dvs. fra tastaturet) er funktionen i almindelighed mest brugt i forbindelse med programmering. Den indirekte formattering er specielt anvendelig sammen med [F], hvis nøjagtighed bestemmes af det valgte antal udlæste decimaler (som forklaret i afsnit 14).

Der er, som sædvanlig, visse begrænsninger, man skal holde sig for øje. De skal f.eks. tænke på, at udlæsningsformatet alene bestemmer, hvor mange cifre der bliver udlæst. I lyspanelregistret som sådan bliver der altid arbejdet med 10 betydende cifre og en 2-cifret eksponent.

Heltalsværdien af tallet i indeksregistret angiver antallet af decimaler efter afrunding. Et tal mindre end 0 fortolkes som 0 (dvs. giver 0 decimaler med [FIX]), medens et tal større end 9 fortolkes som 9 (dvs. giver 9 decimaler med [FIX]).*

* Bemærk, at med [SCI] og [ENG] kan der højst udlæses 7 cifre i mantissen. Alligevel vil et format større end 6 (op til og med 9) påvirke det udlæste tal (se nærmere side 57-58).

⌘ udgør en undtagelse, idet formatteringstallet i R_1 kan ligge i intervallet fra -6 til $+9$. (Dette omtales nærmere i tillæg E, side 244.) Et tal som er mindre end 0 vil ikke påvirke udlæsningsformatet, men alene beregningsnøjagtigheden for funktionen.

Kapitel III
HP-15C
Avancerede funktioner

Afsnit 11

Beregninger med komplekse tal

HP-15C sætter Dem i stand til at arbejde med komplekse tal, dvs. tal på formen

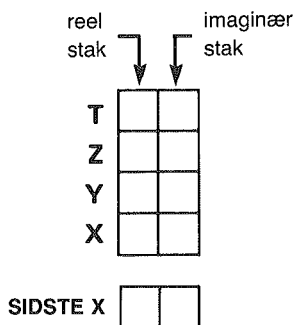
$$a + ib,$$

hvor a = det komplekse tals reeldel,
 b = det komplekse tals imaginærdel og
 $i = \sqrt{-1}$.

Som De straks vil se, er det helt ekstraordinært let at arbejde med komplekse tal på HP-15C. Når først de komplekse tal er lagret, udføres de fleste aritmetiske operationer, som om tallene var reelle.

Den komplekse stak og kompleksstatus

Beregninger med komplekse tal udføres med en kompleksstak, som består af to parallelle registerstakke med hver 4 registre (og to SIDSTE X-registre). Den ene af disse stakke – *reelstakken* – indeholder de komplekse tals reeldele (det er samme stak som bruges under almindelige beregninger). Den anden stak – *imaginærstakken* – indeholder de komplekse tals imaginærdele.



Oprettelse af den komplekse stak

Imaginærstakken oprettes (ved at konvertere fem dataregistre, som beskrevet i tillæg C) når HP-15C sættes i kompleksstatus; den eksisterer ikke uden for kompleksstatus.

Kompleksstatus aktiveres

- 1) automatisk, når $\boxed{f}$ $\boxed{I}$ eller $\boxed{f}$ $\boxed{\text{Re}\geq\text{Im}}$ bruges; eller
- 2) når flag 8, flaget for signalering af kompleksstatus, sættes.

Når HP-15C er i kompleksstatus, vil **C** blive tændt i lyspanelets symbolrække. Det viser, at flag 8 er sat, og at imaginærstakken er oprettet. Når De skifter frem og tilbage mellem normal- og kompleksstatus, vil tallet i lyspanelet være tallet i det reelle *X*-register.

Bemærk: I kompleksstatus (som markeres af **C**-indikatoren) vil HP-15C udføre *alle* trigonometriske funktioner med radian som vinkelmål. Vinkelmålsindikatorerne (**RAD**, **GRAD** eller ingenting – for grader) har kun betydning for to funktioner: $\boxed{\Rightarrow R}$ og $\boxed{\Rightarrow P}$ (dette forhold omtales nærmere i slutningen af dette afsnit).

Deaktivering af kompleksstatus

Da der optages fem dataregistre til imaginærstakken, vil De have mere plads til rådighed for datalagring, programmering og andre avancerede funktioner, hvis De ophæver kompleksstatus, når De udelukkende arbejder med reelle tal.

Kompleksstatus ophæves eller deaktiveres ved at slette flag 8 ($\boxed{g}$ $\boxed{CF}$ 8), hvorved **C** slettes.

Kompleksstatus deaktiveres også, når den kontinuerlige hukommelse nulstilles (se side 62). I begge tilfælde nedbrydes den komplekse stak, og de lagrede tal tabes.

Komplekse tal og stakken

Indtastning af komplekse tal

Et tal med såvel reeldel som imaginærdel indtastes således:

1. Indtast tallets reeldel.
2. Tryk på $\boxed{\text{ENTER}}$.
3. Indtast tallets imaginærdel.
4. Tryk $\boxed{f}$ $\boxed{I}$ (hvis maskinen ikke allerede er i kompleksstatus, vil den automatisk skifte til kompleksstatus og tænde **C**-indikatoren).

Eksempel: Læg $2 + 3i$ til $4 + 5i$. (Operationerne beskrives i diagrammerne, som følger efter indtastningsoversigten.)

Indtastning	Lyspanel	
$\boxed{f}$ $\boxed{FIX}$ 4 2 $\boxed{ENTER}$	2,0000	Indtast reeldelen af første tal i (reelt) Y.
3	3	Indtast imaginærdelen af første tal i (reelt) X.
$\boxed{f}$ $\boxed{I}$	2,0000	Opretter imaginærstak; 3 flyttes over i imaginærstakens X-register medens 2 falder ned i reelstakkens X-register.
4 $\boxed{ENTER}$	4,0000	Indtast reeldelen af det andet tal i (reelt) Y.
5	5	Indtast imaginærdelen af det andet tal i (reelt) X.
$\boxed{f}$ $\boxed{I}$	4,0000	Flytter 5 fra reelt X til imaginært X og flytter 4 fra reelt Y til reelt X.
$\boxed{+}$	6,0000	Summens reeldel.
$\boxed{f}$ $\boxed{(i)}$ (hold nede) (slip)	8,0000 6,0000	Udlæsning af imaginærdelen medens $\boxed{(i)}$ holdes nedtrykket. (Afslutter desuden talindtastning.)

I nedenstående diagrammer kan De se, hvad der sker i henholdsvis reel- og imaginærstakken under hele operationen. (Antag, at stakregistrene allerede indeholder de viste tal.) Bemærk, at imaginærstakken først oprettes, når De indtaster $\boxed{f}$ $\boxed{I}$, og at vi benytter samme konvention som tidligere: registre, hvis indhold er i farezonen, bliver vist med skravering.

	Re	Im	Re	Im	Re	Im	Re	Im	Re	Im
T	9		8		7		7		7	0
Z	8		7		6		6		7	0
Y	7		6		2		2		6	0
X	6		2		2		3		2	3

Taster: 2 ENTER 3 f I

Udførelsen af f I får hele stakken til at falde, T-registrets indhold duplikeres og tallet i det reelle X-register overføres til det imaginære X-register.

Når det andet komplekse tal indtastes, reagerer stakkene som vist nedenfor. Bemærk, at ENTER løfter begge stakke.

	Re	Im	Re	Im	Re	Im	Re	Im
T	7	0	7	0	6	0	6	0
Z	7	0	6	0	2	3	2	3
Y	6	0	2	3	4	0	4	0
X	2	3	4	0	4	0	5	0

Taster: 4 ENTER 5

	Re	Im	Re	Im	Re	Im
T	6	0	6	0	6	0
Z	2	3	6	0	6	0
Y	4	0	2	3	6	0
X	5	0	4	5	6	8

Taster: f I +

Komplekse tal kan desuden indtastes ved først at indtaste imaginærdelen og derefter bruge $\text{Re} \geq \text{Im}$ og $\leftarrow$. Denne metode illustreres i delafsnittet Indtastning af komplekse tal med $\leftarrow$, side 125.

Stakløften i kompleksstatus

Stakløften virker på imaginærstakken som på reelstakken (som fungerer på samme måde i og uden for kompleksstatus). *Funktionernes indvirken på stakløften (aktiv, neutral, blokeret) er præcis de samme for de to registerstakke.* (Stakløften diskuteres nærmere i afsnit 3 og i tillæg B.)

Dertil kommer, at *enhver ikke-neutral funktion med undtagelse af  og  sletter det imaginære X-register, når næste tal indtastes.* Det betyder, at det imaginære X-register nulstilles, når som helst et tal indtastes eller hentes fra et dataregister. Se nærmere på ovenstående stakdiagrammer, som blandt andet illustrerer denne effekt. Det er denne effekt, som er hovedansvarlig for korrespondancen mellem reel og kompleks aritmetik.*

Manipulering af den reelle og den komplekse stak

 (*real exchange imaginary*) ombytter tallene i reel- og imaginærstakkens X-registre, således at reeldelen bliver imaginærdel og vice versa. Y-, Z- og T-registrene påvirkes ikke af  .

  aktiverer desuden kompleksstatus, hvis denne ikke allerede er aktiv.

Midlertidig udlæsning af det imaginære X-register. Tryk på   udlæser indholdet af imaginærstakkens X-register, så længe  holdes nedtrykket. Det væsentlige er, at reelstakkens X-register ikke påvirkes af udlæsningen, som udelukkende er midlertidig.

Fortegnsskift

I kompleksstatus vil  alene påvirke tallet i det reelle X-register – tallet i det imaginære X-register skifter ikke fortegn. Det giver Dem mulighed for at skifte fortegn for enten reel- eller imaginærdel uden at påvirke den anden del. Ved indtastning af negativ reel- eller imaginærdel skal De følge de sædvanlige regler for indtastning af negative tal.

Hvis De ønsker at finde den additivt inverse værdi af et komplekst tal (den værdi, der ved addition med tallet giver 0), der allerede er lagret i X-registret, kan De ikke nøjes med at trykke på  (som De ville kunne med et rent reelt tal). I stedet må De følge én af følgende fremgangsmåder:

* Undtagen for  og , som forklaret senere i dette afsnit (side 131).

■ Gange med -1 .

■ Hvis De ikke ønsker at forstyrre resten af stakken, kan De bruge

CHS **f** **Re $\Rightarrow$ Im** **CHS** **f** **Re $\Rightarrow$ Im**.

Hvis De kun skal skifte fortegn for den ene del af et komplekst tal i X-registret, kan det gøres således:

■ Tryk på **CHS** skifter fortegn på *reeldelen*.

■ **f** **Re $\Rightarrow$ Im** **CHS** **f** **Re $\Rightarrow$ Im** skifter fortegn på *imaginærdelen* og danner således det oprindelige tals *konjugerede*.

Sletning af et komplekst tal

Det er helt sikkert, at De vil få behov for at slette et komplekst tal. De kan kun slette en del ad gangen, men derefter kan De overskrive begge dele (da **◀** og **CLx** blokerer stakløften).

Nulstilling af det reelle X-register. Tryk på **◀** (eller **g** **CLx**) vil i kompleksstatus alene slette tallet i det reelle X-register og sletter altså ikke tallet i det imaginære X-register.

Eksempel: $6 + 8i$ skal ændres til $7 + 8i$ og trækkes fra det foregående indtastede tal. (Brug **f** **Re $\Rightarrow$ Im** eller **f** **(i)** til udlæsning af den imaginære del i X.) Lad a , b , c og d være reelle og komplekse taldele.

	Re	Im	Re	Im	Re	Im	Re	Im
T	a	b	a	b	a	b	a	b
Z	c	d	c	d	c	d	a	b
Y	6	0	6	0	6	0	c	d
X	6	8	0	8	7	8	-1	-8

Taster:



7



(eller en anden operation)

Da nulstilling blokerer stakløften (som nævnt ovenfor), vil det næste tal, som indtastes, overskrive den slettede værdi. Hvis De ønsker at erstatte reeldelen med 0, skal De efter nulstillingen bruge **ENTER** eller en vilkårlig anden funktion til at afslutte talindtastning (ellers vil det næste tal, der indtastes, overskrive nullet). Den imaginære del berøres ikke af denne operation. De kan derefter fortsætte med en vilkårlig anden funktion.

Sletning af det imaginære X-register. Tallet i det imaginære X-register slettes med $\boxed{f} \boxed{\text{Re} \geq \text{Im}}$ efterfulgt af $\boxed{\leftarrow}$. Tryk $\boxed{f} \boxed{\text{Re} \geq \text{Im}}$ igen, så nullet eller det indtastede tal lagres i det imaginære X-register.

Eksempel: Erstat $-1 - 8i$ med $-1 + 5i$.

	Re	Im	Re	Im	Re	Im	Re	Im	Re	Im
T	a	b	a	b	a	b	a	b	a	b
Z	c	d	c	d	c	d	c	d	c	d
Y	e	f	e	f	e	f	e	f	e	f
X	-1	-8	-8	-1	0	-1	5	-1	-1	5

Taster: $\boxed{f} \boxed{\text{Re} \geq \text{Im}}$ $\boxed{\leftarrow}$ 5 $\boxed{f} \boxed{\text{Re} \geq \text{Im}}$
(vilkårlig operation)

Sletning af både det reelle og det imaginære X-register. Hvis De ønsker at slette eller erstatte tallene i både det reelle og det imaginære X-register, skal De blot trykke på $\boxed{\leftarrow}$, som blokerer stakken, hvorefter De kan indtaste det rigtige tal. (Indtast 0 hvis X-registret skal være nulstillet.) Hvis det nye tal skal være reelt (herunder $0 + 0i$), kan De let slette eller erstatte det gamle, komplekse tal ved at trykke på $\boxed{R \downarrow}$ efterfulgt af 0 eller det nye, reelle tal.

Eksempel: Erstat $-1 + 5i$ med $4 + 7i$.

	Re	Im	Re	Im	Re	Im	Re	Im	Re	Im
T	a	b	a	b	c	d	c	d	c	d
Z	c	d	c	d	e	f	e	f	c	d
Y	e	f	e	f	4	5	4	5	e	f
X	-1	5	0	5	4	5	7	0	4	7

Taster: $\boxed{\leftarrow}$ 4 $\boxed{\text{ENTER}}$ 7 $\boxed{f} \boxed{I}$
(vilkårlig operation)

Indtastning af komplekse tal med . Slettefunktionerne og kan bruges sammen med som en alternativ metode til indtastning (og sletning) af komplekse tal. Med denne metode kan De indtaste et komplekstal ved alene at bruge X-registrene – uden at påvirke de øvrige stakregistre. (Dette er muligt fordi og blokerer stakløften.) Udførelse af vil desuden oprette den imaginære stak, hvis den ikke allerede eksisterer.

Eksempel: Indtast $9 + 8i$ uden at bevæge stakken og kvadrér tallet.

Indtastning

Lyspanel

(0,0000)

Forhindrer stakløft, når det næste ciffer (8) indtastes. Undlad dette trin, hvis De ønsker at bevare X-registrets indhold mod at miste T-registrets.

8

8

Først indtastes imaginærdelelen.

7,0000

Udlæsning af foregående tals imaginærdele og aktivering af kompleksstatus.

0,0000

Blokerer stakløften (som er aktiveret af).

9

9

Indtastning af reeldelen (talindtastning afsluttes ikke).

17,0000

Reeldelen.

(hold)
(slip)

144,0000
17,0000

Imaginærdelelen.

	Re	Im
T	a	b
Z	c	d
Y	e	f
X	4	7

	Re	Im
	a	b
	c	d
	e	f
	0	7

	Re	Im
	a	b
	c	d
	e	f
	8	7

	Re	Im
	a	b
	c	d
	e	f
	7	8

Taster:

8

	Re	Im		Re	Im		Re	Im		Re	Im
T	<i>a</i>	<i>b</i>		<i>a</i>	<i>b</i>		<i>a</i>	<i>b</i>		<i>a</i>	<i>b</i>
Z	<i>c</i>	<i>d</i>		<i>c</i>	<i>d</i>		<i>c</i>	<i>d</i>		<i>c</i>	<i>d</i>
Y	<i>e</i>	<i>f</i>		<i>e</i>	<i>f</i>		<i>e</i>	<i>f</i>		<i>e</i>	<i>f</i>
X	7	8		0	8		9	8		17	144

Taster:



9



Indtastning af et reelt tal

De har allerede set to måder, hvorpå et komplekst tal kan indtastes. Reelle tal kan indlæses på en simplere måde: De skal blot indtaste (eller fremkalde) tallet på samme måde som hvis maskinen ikke var i kompleks-status. Når De gør det, vil det imaginære X-register automatisk blive nulstillet (hvis ikke den foregående operation var eller , som forklaret side 122).

På nedenstående stakdiagram kan De se, hvad der sker i stakkene under denne operation. (Antag at den sidst brugte funktion ikke var eller , og at kompleksstakken er uændret fra sidste eksempel.)

	Re	Im		Re	Im		Re	Im
T	<i>a</i>	<i>b</i>		<i>c</i>	<i>d</i>		<i>e</i>	<i>f</i>
Z	<i>c</i>	<i>d</i>		<i>e</i>	<i>f</i>		17	144
Y	<i>e</i>	<i>f</i>		17	144		4	0
X	17	144		4	0		4	0

Taster:

4



(Efterfulgt af et vilkårligt tal.)

Indtastning af et imaginært tal

Et imaginært tal kan indlæses på en ganske let måde, hvis maskinen allerede er i kompleksstatus: De skal blot indtaste eller fremkalde tallet og trykke på $\boxed{f} \boxed{\text{Re} \approx \text{Im}}$.

Eksempel: Indtast $0 + 10i$ (det forudsættes, at den sidste funktion ikke var $\boxed{\leftarrow}$ eller $\boxed{\text{CLx}}$).

Indtastning

10

$\boxed{f} \boxed{\text{Re} \approx \text{Im}}$

Lyspanel

10

0,0000

10 indtastes i det udlæste, reelle X-register og samtidig nulstilles det imaginære X. Ombytning af det reelle og det imaginære X-register. Nu er reeldelen 0, så tallet er rent imaginært.

På nedenstående stakdiagram kan De se, hvad der sker i stakkene under denne operation. (Antag at den sidst brugte funktion ikke var $\boxed{\leftarrow}$ eller $\boxed{\text{CLx}}$, og at kompleksstakken er uændret fra sidste eksempel.)

	Re	Im		Re	Im		Re	Im
T	e	f		e	f		e	f
Z	17	144		17	144		17	144
Y	4	0		4	0		4	0
X	4	0		10	0		0	10

Taster:

10

$\boxed{f} \boxed{\text{Re} \approx \text{Im}}$

(Fortsæt med en vilkårlig operation.)

Bemærk, at brug af $\boxed{f} \boxed{\text{Re} \approx \text{Im}}$ simpelt hen ombytter tallene i X-registre, medens de øvrige stakregistre ikke berøres.

Lagring og fremkald af komplekse tal

STO og **RCL** virker alene på det reelle X-register; så imaginærdelen af et komplekst tal skal lagres og fremkaldes separat. De nødvendige indtastninger kan lagres som del af et program og udføres automatisk.*

De kan lagre det komplekse tal $a + ib$ i registrene R_1 og R_2 således:

STO 1 **f** **Re $\approx$ Im** **STO** 2.

Hvis De derefter bruger **f** **Re $\approx$ Im**, vil den reelle stak blive udlæst. Det komplekse tal $a + ib$ fremkaldes fra dataregistrene R_1 og R_2 på følgende måde:

RCL 1 **RCL** 2 **f** **I**.

Hvis De vil undgå at forstyrre de øvrige stakregistre, kan De i stedet fremkalde tallet med:

RCL 2 **f** **Re $\approx$ Im** **◀** **RCL** 1.

(I programmeringsstatus skal De bruge **g** **CLx** i stedet for **◀**.)

Operationer med komplekse tal

Næsten alle funktioner, som udføres på *reelle tal*, giver ens resultater i og uden for kompleksstatus,** *hvis resultatet vel at mærke er reelt*. Det betyder med andre ord, at valg af kompleksstatus ikke har indflydelse på behandlingen af reelle tal.

Enhver funktion som ikke omtales specielt nedenfor eller i resten af dette afsnit (Beregninger med komplekse tal) ignorerer imaginærstakken.

* De kan bruge HP-15C's matrixfunktioner, som omtales i afsnit 12, til at forenkle lagring og fremkald af komplekse tal. Ved at dimensionere en matrix som $n \times 2$ kan n komplekse tal lagres som rækker i matricen. (Denne teknik er illustreret i *HP-15C Advanced Functions Handbook*, afsnit 3 under *Applications*.)

** Undtagelserne er **►P** og **►R**, som fungerer anderledes i kompleksstatus end udenfor, således at komplekse tal på en enkel måde kan udtrykkes i polært format (side 131).

Funktioner af én variabel

De følgende funktioner virker på såvel reel- som imaginærdelen; dvs. tager det komplekse tal i X-registrene som argument og lagrer funktionsværdien som et komplekst tal i X-registrene:

$\sqrt{x}$ x^2 LN LOG $1/x$ 10^x e^x ABS $\Rightarrow R$ $\Rightarrow P$

Dertil kommer alle trigonometriske og hyperbolske funktioner.*

ABS bestemmer tallets *modulus* eller *numeriske værdi*, der er defineret som kvadratroden af kvadratsummen af reel- og imaginærdelen. Modulus er altid et reelt tal.

$\Rightarrow P$ konverterer et komplekst tal til polær form (produktform), medens $\Rightarrow R$ rekonverterer det til normalformen (sumform). (Se side 131).

Ved brug af de trigonometriske funktioner bliver *radian* anvendt som vinkelmål uanset hvilket vinkelmål der i øvrigt er valgt. Hvis et vinkelmål er opgivet i grader, skal De således bruge $\Rightarrow RAD$, så vinklen omregnes til radian.

Funktioner af to variable

De følgende funktioner virker på såvel reel- som imaginærdelen; dvs. tager de komplekse tal i X- og Y-registrene som argumenter og lagrer funktionsværdien som et komplekst tal i X-registrene. Begge stakke falder, på præcis samme måde som den almindelige rullestak falder, når der arbejdes med reelle tal.

$+$ $-$ $\times$ $\div$ y^x

Funktioner til manipulering af stakken

Når maskinen er i kompleksstatus vil følgende funktioner ændre på reel- og imaginærstakregistrenes indhold på samme måde som uden for kompleksstatus. F.eks. vil $x \rightleftharpoons y$ ombytte X- og Y-registrene i såvel reel- som imaginærstakken.

$x \rightleftharpoons y$ $R \downarrow$ $R \uparrow$ ENTER LSTx

* Se *HP-15C Advanced Functions Handbook*, som definerer de komplekse trigonometriske funktioner og giver yderligere oplysninger om funktioner med komplekse argumenter.

Sammenligningsoperationer

Fire af sammenligningsoperationerne kan benyttes på komplekse tal: de der tester om tal er identiske eller 0. De komplekse tal er ikke ordnede, så det giver ikke mening at benytte f.eks. $\boxed{x \leq y}$.

De kan bruge $\boxed{x=0}$ og $\boxed{\text{TEST}} \ 0$ til at teste om tallet i X-registrene er $0 + 0i$ og De kan tilsvarende bruge $\boxed{\text{TEST}} \ 5$ og $\boxed{\text{TEST}} \ 6$ til at teste tallene i de reelle og imaginære X- og Y-registre for identitet.

$$\boxed{x=0} \quad \boxed{\text{TEST}} \ 0 \ (x \neq 0) \quad \boxed{\text{TEST}} \ 5 \ (x = y) \quad \boxed{\text{TEST}} \ 6 \ (x \neq y)$$

Eksempel: Kompleks aritmetik. Den karakteristiske impedans for et stigenetværk er givet af et udtryk på formen

$$Z_0 = \sqrt{\frac{A}{B}},$$

hvor A og B er komplekse tal. Find Z_0 for (de hypotetiske værdier)
 $A = 1,2 + 4,7i$ og $B = 2,7 + 3,2i$.

Indtastning **Lyspanel**

1.2 $\boxed{\text{ENTER}}$ 4.7 $\boxed{f}$ $\boxed{I}$ **1,2000**

A lagres i reel- og imaginærstakkens X-registre.

2.7 $\boxed{\text{ENTER}}$ 3.2 $\boxed{f}$ $\boxed{I}$ **2,7000**

B lagres i reel- og imaginærstakkens X-registre, hvorved A skubbes op i Y-registrene.

$\boxed{\div}$ **1,0428**

Beregning af A/B .

$\boxed{\sqrt{x}}$ **1,0491**

Beregning af Z_0 og udlæsning af reeldel.

$\boxed{f}$ $\boxed{(I)}$ (hold) **0,2406**

Udlæsning af imaginærdel af Z_0 medens $\boxed{(I)}$ holdes nedtrykket.

(slip) **1,0491**

Reeldelen af Z_0 udlæses igen.

Komplekse resultater af operationer på reelle tal

I de foregående eksempler har indtastningen af komplekse argumenter sikret, at der har været valgt kompleksstatus, når resultatet skulle udlæses. Men der kan optræde situationer, hvor De alene skal bruge kompleksstatus i forbindelse med resultatet – f.eks. ved beregning af $\sqrt{-5}$.

(Uden for kompleksstatus vil en sådan operation medføre fejlmeddelelsen **Error 0** – ugyldig matematisk operation.) I sådanne tilfælde kan De, *uden at forstyrre reelstakkens indhold*, aktivere kompleksstatus ved at sætte flag 8 før De udfører beregningen.*

Eksempel: Arcsin ($\boxed{\boxed{\text{SIN}^{-1}}}$) af 2,404 vil normalt resultere i **Error 0**. Men hvis De udfører operationen i kompleksstatus, vil De få beregnet den komplekse værdi (som ikke er den principale, når argumentet er større end 1) af arcsin.

Indtastning

Lyspanel

$\boxed{\text{g}}$ $\boxed{\text{SF}}$ 8

$\boxed{\text{g}}$ $\boxed{\text{SIN}^{-1}}$

$\boxed{\text{f}}$ $\boxed{(\text{i})}$ (hold)

(slippes)

1,5708

– 1,5239

1,5708

Aktiverer kompleksstatus.

Reeldelen af arcsin 2,404.

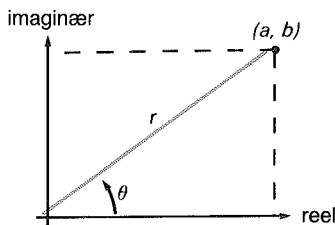
Imaginærdelen af arcsin 2,404.

Reeldelen udlæses igen.

Omregning mellem polære og rektangulære koordinater

I mange anvendelser bliver komplekse tal opgivet polært (på eksponent-, produkt- eller faseform). HP-15C vil dog altid betragte komplekse tal på *sumform*, dvs. som givet i den *komplekse* eller *gaussiske* talplan. De må derfor omregne ethvert tal som er givet polært til sumform, som herefter omtales som *normalformen*.

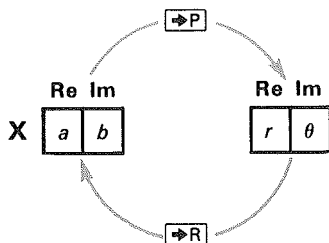
$$a + ib = \begin{cases} r(\cos \theta + i \sin \theta) = re^{i\theta} & \text{(eksponentform)} \\ r \angle \theta & \text{(produkt- eller faseform)} \end{cases}$$



* To tryk på $\boxed{\text{f}}$ $\boxed{\text{Re} \geq \text{Im}}$ udfører samme operation. Sekvensen $\boxed{\text{f}}$ $\boxed{\text{I}}$ benyttes ikke, fordi den vil kombinere tallene i de reelle X- og Y-registre til et enkelt komplekst tal.

$\Rightarrow P$ og $\Rightarrow R$ kan bruges til at omregne mellem de forskellige kompleks-formater. I *kompleksstatus* virker disse funktioner således:

- f $\Rightarrow R$ omregner et tal på polær form til samme tal på normalform ved at erstatte modulus r i det reelle X-register med a og argumentet θ i det imaginære X-register med b .
- g $\Rightarrow P$ omregner et tal på normalform til samme tal på polær form ved at erstatte reeldelen a i X-registret med r og imaginærdelen b i det imaginære X-register med θ .



Disse er de eneste funktioner, der i *kompleksstatus* påvirkes af det aktuelle valg af vinkelmål. Det betyder, at vinkelmålet for θ skal svare til det valgte (som indikeret i symbolrækken).

Eksempel: Bestem summen $2(\cos 65^\circ + i \sin 65^\circ) + 3(\cos 40^\circ + i \sin 40^\circ)$ og udtryk resultatet på polær form. (Med produktformens terminologi skal De bestemme $2 \angle 65^\circ + 3 \angle 40^\circ$.)

Indtastning

g DEG

2 ENTER

65 f I

Lyspanel

2,0000

2,0000

Vælg grader som vinkelmål under omregning mellem polære og rektangulære koordinater.

C-indikatoren tændes; kompleksstatus er aktiv.

Indtastning

 f $\Rightarrow$ R

3 ENTER

40 f I

 f $\Rightarrow$ R

+

 g $\Rightarrow$ P

 f (i) (hold)
(slippes)

Lyspanel
0,8452
3,0000
3,0000
2,2981
3,1434
4,8863
49,9612
4,8863

 Omregning fra polært til rektangulært format. Reel delen (a) udlæses.

 Omregner fra polært til rektangulært format. Reeldelen (a) udlæses.

 Omregner fra rektangulært til polært format. r udlæses. θ (i grader).

Opgaver

Ved at gennemarbejde de følgende opgaver vil De opdage, at det er lige så let at arbejde med komplekse tal som med reelle tal – når man har en HP-15C. I de fleste tilfælde skal der bruges samme indtastninger, når argumenterne er indtastet.

1. Bestem:

$$\frac{2i(-8 + 6i)^3}{(4 - 2\sqrt{5}i)(2 - 4\sqrt{5}i)}$$

Indtastning

 2 f Re $\geq$ Im

8 CHS ENTER

6 f I

 3 y^x
 $\times$

4 ENTER

 5 $\sqrt{x}$

 2 CHS $\times$

f I

+

 2 ENTER 5 $\sqrt{x}$
Lyspanel
0,0000
-8,0000
-8,0000
352,0000
-1.872,0000
4,0000
2,2361
-4,4721
4,0000
-295,4551
2,2361

2i. Reeldelen udlæses.

 $-8 + 6i$.

 $(-8 + 6i)^3$.

 $2i(-8 + 6i)^3$.

 $-2\sqrt{5}$.

 $4 - 2\sqrt{5}i$.

$$\frac{2i(-8 + 6i)^3}{4 - 2\sqrt{5}i}$$

Indtastning

4 CHS ×
f i
÷
f (i)

Lyspanel

- **8,9443****2,0000****9,3982**- **35,1344****9,3982** $2 - 4\sqrt{5}i.$

Reeldel af resultat.

$$\left. \begin{array}{l} 2 - 4\sqrt{5}i. \\ \text{Reeldel af resultat.} \end{array} \right\} 9,3982 - 35,1344i.$$

2. Skriv et program til beregning af udtrykket $\omega = \frac{2z + 1}{5z + 3}$ for forskellige værdier af z . Bestem ω for $z = 1 + 2i$.

(Svar: $0,3902 + 0,0122i$. De kan f.eks. lave følgende program: f
LBL A ENTER ENTER 2 × 1 + x↔y 5 × 3 + ÷ R/S f
Re↔Im g RTN.)

3. Prøv nu at bruge komplekse værdier ved beregning af polynomiet, der blev omtalt side 78. De kan bruge samme program til at bestemme $P(z) = 5z^4 + 2z^3$, hvor z er et komplekst tal.

Prøv først med $z = 7 + 0i$ for at kontrollere, at De får samme resultat som før.

(Svar: $12.691,0000 + 0,0000i$.)

Kør dernæst programmet med $z = 1 + i$.

(Svar: $-24,0000 + 4,0000i$.)

Yderligere information

I *HP-15C Advanced Functions Handbook* kan De finde flere detaljer og tekniske aspekter behandlet. Der gives adskillige illustrative eksempler på anvendelsen af HP-15C til arbejde med komplekse funktioner. Blandt emnerne er:

- Nøjagtighed.
- Vigtige typer af ikke-entydigt bestemte funktioner.
- Komplekse linieintegraler.
- Komplekse potentialer.
- Lagring og fremkald af komplekse tal ved hjælp af en matrix.
- Roduddragning med komplekse tal.
- Løsning af komplekse ligninger.
- Brug af SOLVE og $\int$ i kompleksstatus.

Afsnit 12

Matrixregning

Med HP-15C kan De regne med matricer på en så enkel og effektiv måde, at De kan løse avancerede matematiske problemer med den største let-hed. Maskinen kan arbejde med indtil 5 matricer ad gangen; de navngives A til E, da de adresseres med tasterne $\boxed{A}$ til $\boxed{E}$. HP-15C giver Dem mulighed for at specificere størrelsen af hver enkelt matrix, for at lagre og fremkalde de enkelte elementer, og for at udføre forskellige matrixoperationer på såvel reelle som komplekse matricer. (Sidst i dette afsnit finder De en liste over HP-15C's matrixfunktioner.)

Den vel nok mest udbredte anvendelse af matricer er løsning af lineære ligningssystemer. Vi kan f.eks. betragte ligningerne:

$$3,8x_1 + 7,2x_2 = 16,5$$

$$1,3x_1 - 0,9x_2 = -22,1$$

som skal løses for x_1 og x_2 .

Ligningerne kan på matrixform udtrykkes som $\mathbf{AX} = \mathbf{B}$, hvor

$$\mathbf{A} = \begin{bmatrix} 3.8 & 7.2 \\ 1.3 & -0.9 \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \text{og} \quad \mathbf{B} = \begin{bmatrix} 16.5 \\ -22.1 \end{bmatrix}.$$

De følgende tastsekvenser viser hvor let De kan løse denne matrixopgave med HP-15C. (Matrixoperationerne, der bruges i dette eksempel, bliver gennemgået senere i dette afsnit.)

De skal først dimensionere de to matricer A og B og indtaste de enkelte elementer – række for række, fra venstre mod højre. De skal også definere C som resultatmatrix, dvs. som den matrix, der bruges til at lagre resultatet ($\mathbf{C} = \mathbf{X}$).

Indtastning Lyspanel

[g] [CF] 8
2 [ENTER] [f] [DIM] [A] 2,0000

[f] [MATRIX] 1 2,0000

[f] [USER] **2,0000**
3.8 [STO] [A] A 1.1

3,8000
7.2 [STO] [A] 7,2000
1.3 [STO] [A] 1,3000
.9 [CHS] [STO] [A] - 0,9000
2 [ENTER] 1 [f] [DIM] [B] 1,0000
16.5 [STO] [B] 16,5000
22.1 [CHS] [STO] [B] - 22,1000
[f] [RESULT] [C] - 22,1000

Ophæver kompleksstatus.

Dimensionerer matrix **A** som 2×2 .

Gør klar til automatisk lagring af matrixelementer i USER-status.

Valg af USER-status.

Angiver matrix **A**, række 1, søjle 1. (Dette format anvendes efter indtastning af hvert enkelt matrixelement – indtil De slipper adressetasten (**[A]** - **[E]**)).

3,8 er lagret som $a_{1,1}$.

Lagrer $a_{1,2}$.

Lagrer $a_{2,1}$.

Lagrer $a_{2,2}$.

Dimensionerer **B** som 2×1 .

Lagrer $b_{1,1}$.

Lagrer $b_{2,1}$.

Angiver **C** som resultatmatrix.

Med matrixnotation er løsningen af matrixligningen $\mathbf{AX} = \mathbf{B}$ lig med

$$\mathbf{X} = \mathbf{A}^{-1}\mathbf{B}$$

hvor $\mathbf{A}^{-1}$ er den inverse matrix til **A**. De kan udføre disse operationer ved at indtaste "deskriptorer" for matricerne **B** og **A** i Y- og X-registrene og derefter trykke på **[÷]**. (En deskriptor består af navn og dimension for en matrix.) Bemærk at hvis **A** og **B** var tal, skulle operationen udføres på samme måde.

Indtastning	Lyspanel	
RCL MATRIX B	b 2 1	Udlæsning af deskriptoren for 2×1 konstantmatricen B .
RCL MATRIX A	A 2 2	Udlæsning af deskriptoren for 2×2 koefficientmatricen A . Deskriptoren for B skubbes op i Y-registret.
÷	<i>running</i>	Statusudlæsning medens $A^{-1}B$ bliver beregnet og lagret i C .
	C 2 1	Deskriptor for 2×1 resultatmatricen C .

Frømkald nu elementerne i **C** – dvs. løsningerne til ligningssystemet. (Samtidig skal De ophæve USER-status og slette alle matricer.)

Indtastning	Lyspanel	
RCL C	C 1.1	Viser, at næste udlæste tal er $c_{1,1}$.
	- 11,2887	$c_{1,1} (x_1)$.
RCL C	8,2496	$c_{2,1} (x_2)$.
f USER	8,2496	USER-status ophæves.
f MATRIX 0	8,2496	Alle matricer slettes.

Ligningssystemets løsninger er: $x_1 = -11,2887$ og $x_2 = 8,2496$.

Bemærk: Den følgende gennemgang af matrixoperationer forudsætter, at De allerede har kendskab til teorien bag og brugen af matricer. Mange af funktionerne kan dog anvendes uden større kendskab til den grundlæggende teori.

Matrixdimension

Der kan lagres indtil 64 matricer i hukommelsen. De kan bruge alle 64 elementer i én matrix eller fordele dem blandt indtil fem matricer. F.eks. kan matrixinvertering udføres med en 8×8 matrix med reelle ele-

menter (eller med en 4×4 matrix med komplekse elementer, som beskrevet senere*).

For at spare hukommelse er alle matrixer dimensioneret som 0×0 matrixer, hvis intet er specificeret. Når en matrix dimensioneres eller redimensioneres, bliver det nødvendige antal dataregistre automatisk afsat til matrixen. Det kan være, at De først må ændre splitningen, så der faktisk er dataregistre, som kan afsættes til matrixen. I tillæg C kan De se, hvorledes hukommelsen er organiseret, hvorledes De bestemmer antallet af tilgængelige matrixregistre og hvorledes De kan ændre antallet.

Dimensionering af matrixer

Når De vil dimensionere en matrix til y rækker og x søjler, skal antallet af rækker lagres i Y og antallet af søjler i X. Herefter vil $\boxed{f} \boxed{DIM}$ efterfulgt af en bogstavtast specificere matrixen.

1. Indtast antallet af rækker (y) og tryk på $\boxed{ENTER}$ for at afslutte talindlæsningen.
2. Indtast antallet af søjler (x).
3. Tryk $\boxed{f} \boxed{DIM}$ efterfulgt af en af tastene $\boxed{A} \boxed{E}$ – navnet på matrixen.**

Y	antallet af rækker
X	antallet af søjler

* Matrixfunktionerne som beskrives i dette afsnit virker alene på reelle matrixer. (I kompleks-status bliver den imaginære stak ignoreret under udførelsen af matrixoperationer.) Men HP-15C har fire funktioner, som giver Dem mulighed for at arbejde med reelle repræsentationer af komplekse matrixer – se siderne 158-170.

** Det er ikke nødvendigt at trykke på $\boxed{f}$ før bogstavtasterne – $\boxed{DIM}$ er en gyldig forvalgstast. (Se side 77.)

Eksempel. Dimensionér **A** som en 2×3 matrix.

Indtastning	Lyspanel	
2 ENTER	2,0000	Antallet af rækker lagres i Y-registret.
3	3	Antallet af søjler lagres i X-registret.
f DIM A	3,0000	Matrix A er dimensioneret som 2×3 .

Udlæsning af matrixdimension

De kan få dimensionen af en matrix udlæst på to måder:

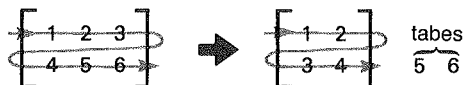
- Tryk **RCL** **MATRIX** efterfulgt af matrixens bogstavtast. Herved bliver navnet udlæst til venstre i lyspanelet, og til højre vises antallet af rækker efterfulgt af antallet af søjler.
- Tryk **RCL** **DIM** efterfulgt af matrixens bogstavtast. Herved bliver antallet af rækker lagret i Y-registret og antallet af søjler i X-registret.

Indtastning	Lyspanel	
RCL MATRIX B	b 0 0	Matrix B har 0 rækker og 0 søjler, da den ikke er særskilt dimensioneret.
RCL DIM A	3,0000	Antal søjler i A .
x$\rightrightarrows$y	2,0000	Antal rækker i A .

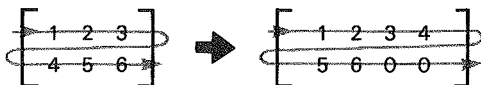
Ændring af matrixdimensionen

De enkelte matrixelementer bliver lagret fortløbende i hukommelsen. Der startes med elementet i første rækkes første søjle, og derefter fra venstre mod højre, række for række indtil hele matricen er lagret ($x_{1,1}$ er altså første element og $x_{n,m}$ er sidste element i $n \times m$ matricen **X**).

Hvis De redimensionerer en matrix til en mindre størrelse, vil De miste nogle af de sidste elementer. Hvis f.eks. 2×3 matricen nederst til venstre redimensioneres til 2×2 , vil to elementer gå tabt:



Hvis De derimod redimensionerer en matrix til en større størrelse, vil alle oprindelige elementer blive bevaret, og alle nye vil automatisk være 0. Hvis De f.eks. redimensionerer samme 2×3 matrix som før til 2×4 , får De:



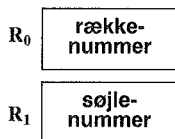
Når De ikke længere skal arbejde med matricer, vil De formodentlig gerne redimensionere alle matricer til 0×0 , så de ikke optager plads i hukommelsen. De kan nuldimensionere alle fem matricer på én gang ved at bruge $\boxed{f} \boxed{MATRIX} 0$. (Matricerne kan nuldimensioneres enkeltvis ved hjælp af $0 \boxed{f} \boxed{DIM} (\boxed{A}, \boxed{B}, \boxed{C}, \boxed{D} \text{ eller } \boxed{E})$.)

Lagring og fremkald af matrixelementer

HP-15C giver Dem mulighed for at lagre og fremkalde matrixelementer på to måder. Med den første metode gennemgås alle elementerne i rækkefølge – fra det første til det sidste. Med den anden metode kan De adressere de enkelte elementer individuelt.

Lagring og fremkald af alle elementer i rækkefølge

HP-15C bruger normalt dataregistre R_0 og R_1 til at lagre række- og søjlenummer for det element, der arbejdes med (en to-dimensional pendant til program-pilen). Hvis maskinen er i USER-status, bliver række- og søjlenumrene automatisk justeret efterhånden som De lagrer/fremkalder de enkelte elementer – fra det første til det sidste.



De kan initialisere række/søjle-tælleren med $\boxed{f} \boxed{MATRIX} 1$, som lagrer 1 i såvel R_0 som R_1 .

Sekventiel lagring eller fremkald af elementer i en given matrix:

1. Sørg for at matricen er rigtigt dimensioneret.
2. Tryk $\boxed{f}$ $\boxed{MATRIX}$ 1. Herved lagres 1 i såvel R_0 som R_1 , således at der startes med første element i matricen.
3. Vælg USER-status ved at trykke $\boxed{f}$ $\boxed{USER}$. Når maskinen er i USER-status, vil række/søjle-tælleren automatisk blive justeret til at indeholde række/søjlenumre for det næste element, når et element er lagret eller fremkaldt.
4. Hvis De lagrer elementer, skal De indtaste det tal, der skal lagres, og derefter trykke $\boxed{STO}$ efterfulgt af matricens bogstavtast.
5. Hvis De fremkalder elementer, skal De trykke på $\boxed{RCL}$ og derefter på matricens bogstavtast.
6. Gentag 4. eller 5. for alle elementer i matricen.

Når De holder matricens bogstavtast nedtrykket efter $\boxed{STO}$ eller $\boxed{RCL}$, bliver matricens navn og række/søjlenumrene for det aktuelle element udlæst. Hvis tasten holdes nedtrykket i mere end 3 sekunder, vil **null** blive udlæst som tegn på, at operationen er blevet annulleret. Det betyder, at elementet ikke lagres/fremkaldes, at R_0 og R_1 ikke ændres, samt endelig at rullestakken er uændret.

Når sidste element er lagret/fremkaldt, skifter række/søjle-tælleren automatisk til første element – dvs. 1 lagres i såvel R_0 som R_1 .

Eksempel: De skal lagre elementerne i nedenstående 2×3 matrix **A**. Husk, at De først skal dimensionere **A**.

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

Indtastning**Lyspanel**

f MATRIX 1

f USER

1 STO A

A 1.1

2 STO A

1,0000**2,0000**

3 STO A

3,0000

4 STO A

4,0000

5 STO A

5,0000

6 STO A

6,0000

RCL A

A 1.1

RCL A

1,0000

RCL A

2,0000

RCL A

3,0000

RCL A

4,0000

RCL A

5,0000

RCL A

6,0000

f USER

6,0000

Lagrer 1 i R_0 og R_1 . (X-registrets indhold bevarer.)

Aktiverer USER-status.

Række 1, søjle 1 i A. (Udlæses i indtil 3 sekunder medens A holdes nedtrykket.)

Værdien af $a_{1,1}$.

Værdien af $a_{1,2}$.

Værdien af $a_{1,3}$.

Værdien af $a_{2,1}$.

Værdien af $a_{2,2}$.

Værdien af $a_{2,3}$.

Fremkald af elementet $a_{1,1}$. (R_0 og R_1 blev etstillet i foregående trin.)

Værdien af $a_{1,1}$.

Værdien af $a_{1,2}$.

Værdien af $a_{1,3}$.

Værdien af $a_{2,1}$.

Værdien af $a_{2,2}$.

Værdien af $a_{2,3}$.

USER-status ophæves.

Kontrol og ændring af enkelte matrixelementer

HP-15C har to måder til at kontrollere (fremkalde) og ændre (lagre) værdien af et enkelt matrixelement. Med første metode benyttes R_0 og R_1 på samme måde som beskrevet ovenfor – bortset fra at række/søjlenummerne ikke ændres automatisk, idet maskinen ikke skal være i USER-status. Med den anden metode benyttes rullestakken til at specificere række- og søjlenummer.

Brug af R_0 og R_1 . Hvis De vil arbejde med et bestemt matricelement kan De lagre dets rækkenummer i R_0 og dets søjlenummer i R_1 . Disse vil ikke automatisk skifte, når $\boxed{\text{STO}}$ eller $\boxed{\text{RCL}}$ bruges (medmindre De har valgt USER-status).

- $x_{n,m}$ fremkaldes af $\boxed{\text{RCL}}$ efterfulgt af matricens bogstavtast, når De har lagret n i R_0 og m i R_1 .
- $x_{n,m}$ lagres af $\boxed{\text{STO}}$ efterfulgt af matricens bogstavtast, når De har lagret n i R_0 og m i R_1 .

Eksempel: De skal lagre 9 i matrix A's 2. rækkes 3. søjle. (A er matricen fra foregående eksempel.)

Indtastning	Lyspanel	
2 $\boxed{\text{STO}}$ 0	2,0000	Rækkenummeret lagres i R_0 .
3 $\boxed{\text{STO}}$ 1	3,0000	Søjlenummeret lagres i R_1 .
9	9	Indtast det nye element $a_{2,3}$.
$\boxed{\text{STO}}$ $\boxed{\text{A}}$	A 2.3	Række 2, søjle 3 i A.
	9,0000	Værdien af $a_{2,3}$.

Brug af rullestakken. De kan også bruge stakregistrene til at specificere et vilkårligt matricelement, således at De bliver fri for at ændre tallene i R_0 og R_1 .

- Et vilkårligt element fremkaldes ved at De indtaster rækkenummer og søjlenummer (i denne rækkefølge med $\boxed{\text{ENTER}}$ som separator). Derefter vil $\boxed{\text{RCL}}$ $\boxed{9}$ efterfulgt af matricens bogstavtast hente elementet frem til X-registret. (Række- og søjlenumrene slettes samtidig.)
- Et vilkårligt element lagres ved at De indtaster værdien og derefter rækkenummer og søjlenummer (i denne rækkefølge med $\boxed{\text{ENTER}}$ som separator). Derefter vil $\boxed{\text{STO}}$ $\boxed{9}$ efterfulgt af matricens bogstavtast hente elementet frem til X-registret. (Række- og søjlenumrene slettes samtidig og matricelementet udlæses igen.)

Bemærk at De skal bruge den blå $\boxed{9}$ -tast umiddelbart foran taster, der ellers har den gule $\boxed{\text{f}}$ -tast som forvalgstast.

Eksempel: Fremkald elementet i matrix A's 2. rækkes 1. søjle. Brug stak-registrene.

Indtastning

2 ENTER 1

RCL g A

Lyspanel

1

4,0000

Rækkenummeret lagres midlertidigt i Y og søjlenummeret i X.

Værdien af $a_{2,1}$.

Lagring af samme tal i alle matrixens elementer

Når De trykker STO MATRIX efterfulgt af matrixens bogstavtast, bliver tallet i lyspanelregistret lagret i alle matrixens elementer.

Matrixoperationer

Matrixoperationer ligner på mange måder almindelige regneoperationer. Regneoperationerne forudsætter at De opgiver nogle tal, der skal arbejdes med (operanderne), og ofte specificerer De selv et register, hvor resultatet skal lagres. På samme måde skal De specificere en eller to matrixer som operander. En *matrixdeskriptor* bruges til at specificere en bestemt matrix. I mange tilfælde vil De få brug for en ny matrix til lagring af resultatet af operationen. Denne matrix kaldes *resultatmatrixen*.

Da matrixoperationer normalt kræver mange individuelle beregninger, vil **running** lyse i lyspanelet under de fleste matrixoperationer.

Matrixdeskriptorer

Tidligere i dette afsnit så De, at når De trykker RCL MATRIX efterfulgt af en bogstavtast, der specificerer (adresserer) matrixen, bliver matrixens navn udlæst til venstre i lyspanelet og antallet af rækker og søjler til højre i lyspanelet. Matrixnavnet er matrixens *deskriptor*. Matrixdeskriptorer kan bevæges rundt i rullestak og dataregistre, præcis som om de var tal.

Når matrixdeskriptoren er lagret i X-registret, bliver matrixens *aktuelle* dimension udlæst.

De bruger matrixdeskriptorer til at bestemme hvilke matricer der skal indgå i den enkelte matrixoperation – deskriptorer er i en vis forstand blot *variabelnavne*. De matrixoperationer, der omtales i resten af dette afsnit, virker på matricer hvis deskriptor er lagret i X-registret og (for nogle operationers vedkommende) i Y-registret.

To matrixoperationer – beregning af en determinant og løsning af matrixligningen $AX = B$ – indebærer en *LU*-faktorisering af matricen, hvis deskriptor er lagret i X.* Under udlæsning af matrixdeskriptoren bliver en *LU*-faktoriseret matrix indikeret af to bindestreger efter matrixnavnet. (Se side 158, hvor brugen af matricer på *LU*-form bliver gennemgået.)

Resultatmatricen

I forbindelse med mange af de operationer, der omtales i dette afsnit, skal De definere en matrix til lagring af det beregnede resultat. Denne matrix kaldes *resultatmatricen*.

Der er andre matrixoperationer, der ikke kræver definition af en resultatmatrix (hvad der gælder, gøres klart for hver enkelt operation). En sådan operation vil enten erstatte den originale matrix med resultatet af operationen (hvis resultatet er en matrix – som f.eks. ved transponering) eller lagre resultatet i X-registret (hvis resultatet er et tal – som f.eks. række-normen).

Før De udfører en operation, der kræver en resultatmatrix, skal De definere resultatmatricen. Tryk f RESULT og på den bogstavtast, der skal navngive resultatmatricen. (Hvis deskriptoren for den ønskede resultatmatrix bliver udlæst, kan De i stedet trykke STO RESULT.) Den således valgte matrix vil fortsat være resultatmatrix indtil en anden vælges.** Deskriptoren for den valgte resultatmatrix kan udlæses med RCL RESULT.

Når De udfører en operation, der påvirker resultatmatricen, bliver den automatisk redimensioneret til den rette størrelse. Hvis redimensionerin-

* *LU*-faktoriseringen af en matrix **A** er en anden matrix som kan betragtes som en nedre trekantmatrix **L** og en øvre trekantmatrix **U**, hvis produkt **LU** er lig **A** (eventuelt med nogle rækker ombyttet). *HP-15C Advanced Functions Handbook* diskuterer *LU*-faktorisering i detaljer.

** Matrix **A** er automatisk valgt som resultatmatrix, når den kontinuerlige hukommelse nulstilles.

gen kræver flere dataregistre end der er plads til i matrixhukommelsen (højest 64 for alle fem matricer tilsammen), vil operationen ikke blive udført. Denne restriktion kan man ofte komme ud over ved at lade resultatmatricen være en af de matricer, der indgår i operationen. (Der er dog visse operationer, hvor resultatmatricen skal være en selvstændig matrix – dette vil blive særskilt bemærket i forbindelse med disse operationer.)

Når tasten for en operation der lagrer resultatet i en resultatmatrix holdes nedtrykket, vil resultatmatricens deskriptor blive udlæst. Hvis tasten slippes inden der er gået 3 sekunder, bliver operationen udført, og resultatmatricens deskriptor bliver lagret i X-registret. Hvis tasten holdes nedtrykket i længere end 3 sekunder, annulleres operationen, og **null** bliver udlæst.

Kopiering af en matrix

Kopiering af alle elementer fra en matrix over i de samme elementer i en anden matrix kan udføres med **RCL** og **STO** **MATRIX**.

1. Tryk **RCL** **MATRIX** efterfulgt af bogstavtasten for matricen, der skal kopieres. Matricens deskriptor bliver udlæst.
2. Tryk **STO** **MATRIX** efterfulgt af bogstavtasten for den matrix, der skal kopieres over i.

Hvis matricen, hvis navn specificeres efter **STO** **MATRIX**, ikke har samme dimension som originalmatricen, vil den automatisk blive redimensioneret, så den passer.

Eksempel: Kopiér matrix **A** fra det foregående eksempel over i matrix **B**.

Indtastning	Lyspanel	
RCL MATRIX A	A 2 3	Udlæsning af originalmatricens deskriptor.
STO MATRIX B	A 2 3	Redimensionering af B og kopiering af A til B .
RCL MATRIX B	b 2 3	Deskriptoren for B bliver udlæst.

Operationer på én matrix

Følgende oversigt viser, hvilke funktioner der virker på kun én matrix – matricen hvis deskriptor udlæses - altså svarer til funktioner af en varia-

bel. Operationer, der virker på en enkelt matrix samt et tal i et andet stakregister (en skalar), er beskrevet under Skalare operationer side 149.

Indtastning	Resultat i X-register	Effekt på matrixen hvis resultat er lagret i X	Effekt på resultatmatrixen
CHS	Ingen ændring.	Skifter fortegn på alle elementer.	Ingen***
1/x	Deskriptor for resultatmatrix.	Ingen.***	Den specificerede matrices inverse.****
f MATRIX 4	Deskriptor for den transponerede.	Erstattet af den transponerede.	Ingen.***
f MATRIX 7	Den specificerede matrices rækkenorm.*	Ingen.	Ingen.
f MATRIX 8	Den specificerede matrices euklidiske norm.**	Ingen.	Ingen.
f MATRIX 9	Den specificerede matrices determinant.	Ingen.***	LU-faktorisering af den specificerede matrix.****

* Rækkenormen er den største af de absolutte rækkesummer, de summer der dannes ved rækkevis at addere de numeriske værdier.

** Den euklidiske norm er lig kvadratroden af kvadratsummen af alle den specificerede matrices elementer.

*** Medmindre resultatmatrixen er den samme som specificeret i X-registret.

**** Hvis den specificerede matrix er singulær (dvs. Ikke har en Invers matrix) vil HP-15C modificere LU-formen med en størrelse, der normalt er lille i forhold til afrundingsfejlen. Med **1/x** er den beregnede inverse den inverse til en matrix, der er meget lig den originale, singulære matrix. (Se *HP-15C Advanced Functions Handbook* for yderligere information.)

Eksempel: Beregn den transponerede af matrix **B**, som i det foregående eksempel blev defineret som:

$$\mathbf{B} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 9 \end{bmatrix}.$$

Indtastning

RCL **MATRIX** **B**

f **MATRIX** 4

Lyspanel

b **2** **3**

b **3** **2**

Udlæsning af deskriptoren for 2×3 matricen **B**.

Deskriptor for den transponerede.

Matrix **B** (som kan udlæses med **RCL** **B** i USER-status) er nu:

$$\mathbf{B} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 9 \end{bmatrix}.$$

Skalare operationer

Skalare operationer udfører aritmetiske operationer mellem en skalar (dvs. et rent tal) og hvert enkelt element i en matrix. Skalaren og matrixens deskriptor skal være lagret i X- og Y-registrene. Bemærk, at det har betydning for udfaldet af **+** og **-** om skalaren er lagret i X eller Y. De resulterende værdier lagres i de korresponderende resultatmatrixelementer.

Tabellen på næste side opregner de mulige skalare operationer.

Operation	Elementer i resultatmatricen*	
	Matrix i Y-register Skalar i X-register	Skalar i Y-register Matrix i X-register
$\boxed{+}$	Adderer den skalare værdi til hvert enkelt matrixelement.	
$\boxed{\times}$	Multipliserer hvert enkelt matrixelement med den skalare værdi.	
$\boxed{-}$	Trækker den skalare værdi fra hvert matrixelement.	Trækker hvert matrixelement fra den skalare værdi.
$\boxed{\div}$	Dividerer hvert matrixelement med den skalare værdi.	Beregner matrixens inverse og ganger hvert matrixelement med den skalare værdi.
* Resultatmatricen kan være den specificerede matrix.		

Eksempel: Beregn matricen $\mathbf{B} = 2\mathbf{A}$ og træk derefter 1 fra hvert element i $\mathbf{B}$. Brug $\mathbf{A}$ fra før:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 9 \end{bmatrix}.$$

Indtastning

$\boxed{f}$ $\boxed{\text{RESULT}}$ $\boxed{\mathbf{B}}$

$\boxed{\text{RCL}}$ $\boxed{\text{MATRIX}}$ $\boxed{\mathbf{A}}$

2 $\boxed{\times}$

Lyspanel

$\mathbf{A}$ 2 3

$\mathbf{B}$ 2 3

$\mathbf{B}$ bruges som resultatmatrix.

Deskriptor for $\mathbf{A}$.

Redimensionerer $\mathbf{B}$ til samme dimension som $\mathbf{A}$, ganger $\mathbf{A}$'s elementer med 2 og lagrer resultaterne i de korresponderende elementer i $\mathbf{B}$ og udlæser endelig deskriptoren for resultatmatricen $\mathbf{B}$.

Indtastning1 **Lyspanel****b** **2** **3**

Trækker 1 fra hvert element
i **B** som erstattes af resultatet.

Resultatet, som kan udlæses med i USER-status, er:

$$\mathbf{B} = \begin{bmatrix} 1 & 3 & 5 \\ 7 & 9 & 17 \end{bmatrix}.$$

Aritmetiske operationer

Med matrixdeskriptorer i såvel X som Y vil og beregne matrixsum og -different.

Indtastning	Beregning*
+	$\mathbf{Y} + \mathbf{X}$
-	$\mathbf{Y} - \mathbf{X}$
* Resultatet lagres i resultatmatricen, som kan være X eller Y.	

Eksempel: Beregn $\mathbf{C} = \mathbf{B} - \mathbf{A}$, hvor **B** og **A** er defineret i foregående eksempel.

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 9 \end{bmatrix} \quad \text{og} \quad \mathbf{B} = \begin{bmatrix} 1 & 3 & 5 \\ 7 & 9 & 17 \end{bmatrix}.$$

Indtastning**Lyspanel****b** **2** **3**

C er resultatmatrix.

B's deskriptor udlæses.
(Dette trin kan udelades,
hvis deskriptoren allerede
udlæses.)

A **2** **3**

Fremkald af deskriptoren for
A. Deskriptoren for **B** rykkes
samtidig op i Y.

Indtastning**Lyspanel****C 2 3**Beregning af $\mathbf{B} - \mathbf{A}$ med lagring af resultatet i $\mathbf{C}$.

Resultatet er

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 8 \end{bmatrix}$$

Matrixmultiplikation

Med matrixdeskriptorer i både $\mathbf{X}$ og $\mathbf{Y}$ kan De beregne hele tre forskellige matrixprodukter. Skemaet nedenfor viser resultatet af hver af de tre funktioner, når matrix $\mathbf{X}$ er specificeret i $\mathbf{X}$ -registret og matrix $\mathbf{Y}$ er specificeret i $\mathbf{Y}$ -registret. Matricen $\mathbf{X}^{-1}$ er den inverse af $\mathbf{X}$ og $\mathbf{Y}^T$ er $\mathbf{Y}$'s transponerede.

Indtastning	Beregning*
	$\mathbf{Y} \mathbf{X}$
5	$\mathbf{Y}^T \mathbf{X}$
	$\mathbf{X}^{-1} \mathbf{Y}$
* Resultatet lagres i resultatmatricen. I tilfældet kan $\mathbf{Y}$ men ikke $\mathbf{X}$ være resultatmatrix. I de øvrige tilfælde skal der vælges en uafhængig resultatmatrix.	

Bemærk: Når De bruger til at bestemme $\mathbf{A}^{-1} \mathbf{B}$ skal De indlæse deskriptorerne i rækkefølgen $\mathbf{B}$, $\mathbf{A}^{-1}$ – og ikke som de optræder i formlen.*

De resulterende værdier svarer ganske til dem, der opnås ved at udføre beregningerne efter de sædvanlige regler for matrixmultiplikation.

5 ændrer ikke den matrix, som er specificeret i $\mathbf{Y}$, selv om dennes transponerede bruges. Resultatet er det samme, som hvis man havde brugt 4 og .

* Det er samme indtastningsrækkefølge som skal bruges ved beregning af $a^{-1}b = b/a$.

Når $\boxed{+}$ bruges bliver matricen, der er specificeret i X , erstattet af dens LU -faktorisering. $\boxed{\div}$ beregner $X^{-1}Y$ ved hjælp af en mere direkte metode end $\boxed{1/x}$ og $\boxed{\times}$, således at beregningen bliver hurtigere og mere nøjagtig.

Eksempel: Brug matricerne A og B fra det foregående eksempel til at beregne $C = A^T B$.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 9 \end{bmatrix} \quad \text{og} \quad B = \begin{bmatrix} 1 & 3 & 5 \\ 7 & 9 & 17 \end{bmatrix}$$

Indtastning

$\boxed{\text{RCL}}$ $\boxed{\text{MATRIX}}$ $\boxed{A}$

Lyspanel

A **2** **3**

Fremkalder deskriptoren for A .

$\boxed{\text{RCL}}$ $\boxed{\text{MATRIX}}$ $\boxed{B}$

b **2** **3**

Lagrer deskriptoren for B i X . Deskriptoren for A rykker op i Y .

$\boxed{f}$ $\boxed{\text{RESULT}}$ $\boxed{C}$

b **2** **3**

C bruges som resultatmatrix.

$\boxed{f}$ $\boxed{\text{MATRIX}}$ 5

C **3** **3**

$A^T B$ beregnes og lagres i C , som redimensioneres til 3×3 .

Resultatet, matricen C , er:

$$C = \begin{bmatrix} 29 & 39 & 73 \\ 37 & 51 & 95 \\ 66 & 90 & 168 \end{bmatrix}.$$

Løsning af ligningen $AX = B$

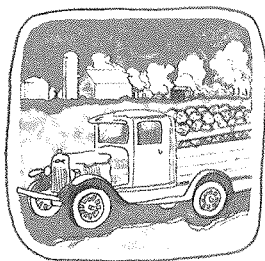
$\boxed{\div}$ er praktisk ved løsning af lineære ligningssystemer på formen $AX = B$, hvor A er koeficientmatricen, B er konstantmatricen og X er resultatmatricen. Deskriptoren for konstantmatricen B skal lagres i Y-registret og deskriptoren for koeficientmatricen A skal lagres i X-registret. Når De trykker på $\boxed{\div}$, bliver $X = A^{-1}B$ beregnet.*

Y	konstant- matrix
X	koeficient- matrix

Husk at $\boxed{\div}$ erstatter koeficientmatricen med dens LU -faktorisering og at denne matrix ikke kan bruges som resultatmatrix. Husk også at $\boxed{\div}$ er hurtigere og nøjagtigere end en løsning med $\boxed{1/x}$ og $\boxed{\times}$.

I begyndelsen af dette afsnit fandt De løsningerne til et lineært ligningssystem, hvor både konstant- og løsningsmatrix var vektorer. I det følgende eksempel vil vi illustrere hvorledes HP-15C kan finde løsninger i beregninger, hvor der er mere end ét sæt konstanter.

Eksempel: Jørgen Bonde har netop modtaget nedenstående kontoudtog for sine tre sidste leverancer af grønkål og broccoli.



* Hvis A er singulær (og derfor ikke har nogen invers matrix) vil HP-15C modificere LU -faktoriseringen med en størrelse, der normalt er lille i forhold til afrundingsfejlen. Den beregnede løsning svarer til løsningen for et ikke-singulært system, som ligger nær det originale.

	Uge		
	1	2	3
Total vægt (kg)	274	233	331
Total værdi (kr)	1.203,20	1.129,60	1.513,60

Jørgen Bonde ved, at han modtager 2,40 kr/kg for sin grønkål og 8,60 kr/kg for broccolien. Men han har forlagt sin opgørelse over, hvor meget han har leveret af hver type kål i de forløbne tre uger. Opgaven kan løses med en matrixligning, hvor der er to søjler i konstantmatricen.

Løsning: Hver uges leverance repræsenterer to lineære ligninger (en for vægt og en for værdi) med to ubekendte (vægt af grønkål og vægt af broccoli). Alle tre uger kan behandles på én gang med matrixligningen:

$$\begin{bmatrix} 1 & 1 \\ 2,4 & 8,6 \end{bmatrix} \begin{bmatrix} d_{1,1} & d_{1,2} & d_{1,3} \\ d_{2,1} & d_{2,2} & d_{2,3} \end{bmatrix} = \begin{bmatrix} 274 & 233 & 331 \\ 1.203,20 & 1.129,60 & 1.513,60 \end{bmatrix}$$

eller

$$\mathbf{AD} = \mathbf{B}$$

hvor første række af matricen **D** er mængden af grønkål uge for uge og anden række er de tilsvarende tal for broccoli.

Indtastning

Lyspanel

2 [ENTER] f [DIM] A **2,0000**

Matrix **A** har dimensionen 2×2 .

f [MATRIX] 1 **2,0000**

Række- og søjlenummer i R_0 og R_1 sættes til 1.

f [USER] **2,0000**

USER-status.

1 [STO] A **1,0000**

Lagring af $a_{1,1}$.

[STO] A **1,0000**

Lagring af $a_{1,2}$.

2.4 [STO] A **2,4000**

Lagring af $a_{2,1}$.

8.6 [STO] A **8,6000**

Lagring af $a_{2,2}$.

2 [ENTER] 3 f [DIM] B **3,0000**

Matrix **B** har dimensionen 2×3 .

Indtastning	Lyspanel	
274 STO B	274,0000	Lagring af $b_{1,1}$.*
233 STO B	233,0000	Lagring af $b_{1,2}$.
331 STO B	331,0000	Lagring af $b_{1,3}$.
1203.2 STO B	1.203,2000	Lagring af $b_{2,1}$.
1129.6 STO B	1.129,6000	Lagring af $b_{2,2}$.
1513.6 STO B	1.513,6000	Lagring af $b_{2,3}$.
f RESULT D	1.513,6000	D bruges som resultatmatrix.
RCL MATRIX B	b 2 3	Konstantmatricens deskriptor.
RCL MATRIX A	A 2 2	Koefficientmatricens deskriptor står nu i X-registret, medens konstantmatricens står i Y-registret.
÷	d 2 3	$A^{-1}B$ beregnes og lagres i D.
RCL D	186,0000	Fremkald af $d_{1,1}$, vægten af grønkål i første uge.
RCL D	141,0000	Fremkald af $d_{1,2}$, vægten af grønkål i anden uge.
RCL D	215,0000	Fremkald af $d_{1,3}$.
RCL D	88,0000	Fremkald af $d_{2,1}$.
RCL D	92,0000	Fremkald af $d_{2,2}$.
RCL D	116,0000	Fremkald af $d_{2,3}$.
f USER	116,0000	USER-status ophæves.

* Bemærk at det ikke er nødvendigt at bruge 1 før De begynder indtastningen af elementerne i matrix B. Det skyldes at række- og søjlenumrene i R_0 og R_1 automatisk bliver sat til 1, når alle elementer i matrix A har været behandlet.

Jørgen Bondes leverancer fordeler sig således:

	Uge		
	1	2	3
Grønkål (kg)	186	141	215
Broccoli (kg)	88	92	116

Beregning af restmatrix

HP-15C giver Dem mulighed for at beregne en restmatrix, der er defineret af

$$\text{Restmatrix} = \mathbf{R} - \mathbf{YX}$$

hvor $\mathbf{R}$ er resultatmatricen og $\mathbf{X}$ og $\mathbf{Y}$ er matricerne, hvis deskriptorer er lagrede i X- og Y-registrene.

Denne mulighed er specielt bekvem, når man f.eks. bruger en iterativ proces til at finpudse løsningen af et sæt lineære ligninger og i forbindelse med lineær regression. Hvis f.eks. $\mathbf{C}$ er en mulig løsning for $\mathbf{AX} = \mathbf{B}$ vil $\mathbf{B} - \mathbf{AC}$ være en indikator for hvor godt løsningen tilfredsstiller ligningen. (Se nærmere i *HP-15C Advanced Functions Handbook*, som indeholder detaljer om finpudsning af iterative processer og om lineær regression.)

Restfunktionen (**MATRIX** 6) anvender det aktuelle indhold af resultatmatricen og de matricer som er specificeret i X og Y til at beregne differensmatricen. Restmatricen lagres under resultatmatricens navn og overskriver den oprindelige resultatmatrix. De kan ikke bruge de matricer, der er specificeret i X eller Y, som resultat/restmatrix.

MATRIX 6 giver bedre resultater end **×** efterfulgt af **-** – specielt hvis restværdierne er små i forhold til de matrixelementer der trækkes fra hinanden.

De kan beregne restmatricen således:

1. Indtast deskriptoren for $\mathbf{Y}$ med **ENTER**, så $\mathbf{Y}$ lagres i Y-registret.
2. Indtast deskriptoren for $\mathbf{X}$.
3. Vælg $\mathbf{R}$ som resultatmatrix.
4. Brug **f** **MATRIX** 6 til at beregne restmatricen, som erstatter den oprindelige resultatmatrix, hvis deskriptor bliver udlæst.

Brug af LU -faktoriserede matricer

Som tidligere omtalt vil to af matrixoperationerne (beregning af determinant og løsning af matrixligningen $\mathbf{AX} = \mathbf{B}$) danne en LU -faktorisering af matricen, hvis deskriptor bliver udlæst. Deskriptoren for en sådan matrix har to bindestreger mellem bogstavet og dimensionsangivelsen. Elementerne i en matrix på LU -form afviger fra elementerne i den oprindelige matrix.

Men alligevel kan deskriptoren for en matrix på LU -form bruges i stedet for deskriptoren for den oprindelige matrix, når der arbejdes med operationer, som indebærer brug af den inverse eller beregning af determinanten. Det betyder, at det for følgende operationer er ligegyldigt, om man bruger den originale eller den LU -faktoriserede form:

1/x
÷ for matricen i X-registret
MATRIX 9

Disse funktioner danner automatisk en LU -faktorisering undervejs og bruger deskriptoren til at afgøre, om faktoriseringen allerede er udført. Resultaterne er således de samme, uanset om De bruger originalmatricen eller den LU -faktoriserede.

Hvis De f.eks. skal løse matrixligningen $\mathbf{AX} = \mathbf{B}$, bliver $\mathbf{A}$ ændret til sin LU -form. Hvis De ønsker at ændre konstantmatricen $\mathbf{B}$ og prøve at løse ligningssystemet igen, kan De gøre dette uden at ændre $\mathbf{A}$ – LU -matricen giver den korrekte løsning.

Alle andre matrixoperationer betragter LU -faktoriseringen som en selvstændig matrix uden forbindelse med den originale. Det betyder, at De ikke kan regne med at opnå korrekte resultater, hvis De bruger LU -faktoriseringen i stedet for originalmatricen.

Beregninger med komplekse matricer

HP-15C giver Dem mulighed for at udføre matrixmultiplikation med og invertering af komplekse matricer (dvs. matricer hvis elementer er komplekse tal) samt at løse komplekse ligningssystemer (dvs. lineære ligningssystemer, hvor koefficienter, konstanter og rødder er komplekse tal).

Men HP-15C kan kun lagre og arbejde på reelle matricer, og det bør understreges, at maskinens muligheder for at behandle komplekse matricer er ganske uafhængig af de stakmekanismer, der blev omtalt under beregninger med komplekse tal.

De behøver ikke skifte til kompleksstatus, når De vil arbejde med komplekse matricer.

I stedet udføres selve beregningerne med reelle matricer, som er afledt af de komplekse – hvorledes det gøres, beskrives nedenfor – og underkastet visse transformationer udover de sædvanlige almindelige matrixoperationer. Disse transformationer udføres med fire specialfunktioner, som vil blive beskrevet i det følgende. (Der bliver givet flere eksempler på brugen af komplekse matricer i *HP-15C Advanced Functions Handbook*.)

Bemærk: I teksten bruges såvel $x_{r,s}$ som x_{rs} til at angive matrixelementer.

Lagring af elementerne i den komplekse matrix

Betragt en $m \times n$ kompleks matrix, $\mathbf{Z} = \mathbf{X} + i\mathbf{Y}$, hvor $\mathbf{X}$ og $\mathbf{Y}$ er reelle $m \times n$ matricer. Denne matrix kan repræsenteres ved hjælp af en partitioneret matrix:

$$\mathbf{Z}^P = \left[\begin{array}{c} \mathbf{X} \\ \mathbf{Y} \end{array} \right] \left. \begin{array}{l} \text{Reelle del} \\ \text{Imaginære del} \end{array} \right\}$$

Det løftede indeks P bruges til at markere, at matricen er partitioneret.

Alle elementer i $\mathbf{Z}^P$ er reelle tal. Elementerne i den øverste halvdel repræsenterer den reelle matrix, $\mathbf{X}$, og elementerne i den nederste halvdel repræsenterer den imaginære matrix, $\mathbf{Y}$. Elementerne i $\mathbf{Z}^P$ kan lagres i enhver af de sædvanlige fem matricer (f.eks. $\mathbf{A}$) på samme måde som tidligere omtalt for reelle matricer.

Hvis vi f.eks. har $\mathbf{Z} = \mathbf{X} + i\mathbf{Y}$, hvor

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} \quad \text{og} \quad \mathbf{Y} = \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix},$$

kan $\mathbf{Z}$ repræsenteres i lommeregneren af

$$\mathbf{A} = \mathbf{Z}^P = \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix}.$$

Antag at De får brug for at regne på en kompleks matrix, som ikke er skrevet som summen af en reel og en imaginær matrix, men hvis elementer derimod alle er givet på kompleks form, som f.eks.

$$\mathbf{Z} = \begin{bmatrix} x_{11} + iy_{11} & x_{12} + iy_{12} \\ x_{21} + iy_{21} & x_{22} + iy_{22} \end{bmatrix}.$$

Denne matrix kan i lommeregneren repræsenteres af en reel matrix, som ligner den uhyre meget – som dannes ved at man ignorerer i og $+$. 2×2 matricen $\mathbf{Z}$ kan på denne måde repræsenteres af en 2×4 matrix.

$$\mathbf{A} = \mathbf{Z}^C = \begin{bmatrix} x_{11} & y_{11} & x_{12} & y_{12} \\ x_{21} & y_{21} & x_{22} & y_{22} \end{bmatrix}.$$

Det løftede indeks C bruges til at markere, at matricen er givet på C -form (kompleksform).

Hvis en matrix er givet på C -form, skal De transformere den til P -form, idet alle matrixoperationerne forudsætter P -form. Til det formål har HP-15C to funktioner, som (blandt andet) omsætter komplekse matricer mellem C - og P -form.

Indtastning	Transformerer	Til
$\boxed{\text{f}} \boxed{\text{Py},x}$	$\mathbf{Z}^C$	$\mathbf{Z}^P$
$\boxed{\text{g}} \boxed{\text{Cy},x}$	$\mathbf{Z}^P$	$\mathbf{Z}^C$

Når en matrix skal transformeres, skal dens deskriptor lagres i X , hvorefter $\boxed{\text{f}} \boxed{\text{Py},x}$ eller $\boxed{\text{g}} \boxed{\text{Cy},x}$ udfører transformationen. Transformationen udføres alene på den specificerede matrix; resultatmatricen berøres ikke.

Eksempel: Følgende komplekse matrix lagres

$$Z = \begin{bmatrix} 4 + 3i & 7 - 2i \\ 1 + 5i & 3 + 8i \end{bmatrix}$$

på C -form. Derefter skal den transformeres til Z^P , så den kan bruges til beregninger.

De kan gøre dette ved at lagre Z^C 's elementer i matrix A og derefter transformere med $[Py,x]$. A bliver således:

$$A = Z^C = \begin{bmatrix} 4 & 3 & 7 & -2 \\ 1 & 5 & 3 & 8 \end{bmatrix}.$$

Indtastning

Lyspanel

[f] [MATRIX] 0					Sletter alle matricer.
2 [ENTER] 4 [f] [DIM] [A]	4,0000				A får dimensionen 2×4 .
[f] [MATRIX] 1	4,0000				Initialiserer R_0 og R_1 .
[f] [USER]	4,0000				Skifter til USER-status.
4 [STO] [A]	4,0000				Lagrer $a_{1,1}$.
3 [STO] [A]	3,0000				Lagrer $a_{1,2}$.
7 [STO] [A]	7,0000				Lagrer $a_{1,3}$.
2 [CHS] [STO] [A]	-2,0000				Lagrer $a_{1,4}$.
1 [STO] [A]	1,0000				Lagrer $a_{2,1}$.
5 [STO] [A]	5,0000				Lagrer $a_{2,2}$.
3 [STO] [A]	3,0000				Lagrer $a_{2,3}$.
8 [STO] [A]	8,0000				Lagrer $a_{2,4}$.
[f] [USER]	8,0000				Ophæver USER-status.
[RCL] [MATRIX] [A]	A	2	4		Udlæser deskriptor for matrix A.
[f] [Py,x]	A	4	2		Transformerer Z^C til Z^P . Samtidig redimensioneres A.

Matrix A repræsenterer nu den komplekse matrix Z på P -form:

$$A = Z^P = \begin{bmatrix} 4 & 7 \\ 1 & 3 \\ \hline 3 & -2 \\ 5 & 8 \end{bmatrix} \quad \left. \begin{array}{l} \} \\ \} \end{array} \right\} \begin{array}{l} \text{Reel del} \\ \text{Imaginær del} \end{array}$$

De komplekse transformationer

Der skal udføres endnu en transformation, hvis De ønsker at multiplicere de komplekse matricer. Det samme gælder for invertering af en kompleks matrix. Disse transformationer konverterer en $m \times n$ kompleks matrix på P -form til en $2m \times 2n$ partitioneret matrix på følgende form:

$$\tilde{\mathbf{Z}} = \begin{bmatrix} \mathbf{X} & -\mathbf{Y} \\ \mathbf{Y} & \mathbf{X} \end{bmatrix}.$$

Matricen $\tilde{\mathbf{Z}}$ dannes af MATRIX 2 og har dobbelt så mange elementer som $\mathbf{Z}^P$.

Matricerne nedenfor viser, hvilken sammenhæng der er mellem matricerne $\tilde{\mathbf{Z}}$ og $\mathbf{Z}^P$.

$$\mathbf{Z}^P = \left[\begin{array}{cc|cc} 1 & -6 & & \\ \hline -4 & 5 & & \end{array} \right] \longleftrightarrow \tilde{\mathbf{Z}} = \left[\begin{array}{cc|cc} 1 & -6 & 4 & -5 \\ \hline -4 & 5 & 1 & -6 \end{array} \right]$$

Nedenstående oversigt viser, hvilke funktioner der bruges til transformation mellem $\tilde{\mathbf{Z}}$ og $\mathbf{Z}^P$.

Indtastning	Transformerer	Til
f MATRIX 2	$\mathbf{Z}^P$	$\tilde{\mathbf{Z}}$
f MATRIX 3	$\tilde{\mathbf{Z}}$	$\mathbf{Z}^P$

Når en matrix skal transformeres, skal dens deskriptor lagres i X, hvorefter f MATRIX 2 eller f MATRIX 3 udfører transformationen. Transformationen udføres alene på den specificerede matrix; resultatmatricen berøres ikke.

Invertering af en kompleks matrix

De kan beregne den inverse af en kompleks matrix ved at udnytte sammenhængen: $(\tilde{\mathbf{Z}})^{-1} = (\mathbf{Z}^{-1})$.

Invertering af en kompleks matrix, $\mathbf{Z}$, udføres således:

1. $\mathbf{Z}$'s elementer lagres i dataregistre – på C - eller P -form.
2. Deskriptoren for $\mathbf{Z}$ lagres i $\mathbf{X}$.
3. Hvis matricen blev lagret på C -form, skal $\boxed{\text{f}} \boxed{\text{Py},x}$ bruges til at transformere til $\mathbf{Z}^P$.
4. Tryk på $\boxed{\text{f}} \boxed{\text{MATRIX}} 2$, så $\mathbf{Z}^P$ transformeres til $\tilde{\mathbf{Z}}$.
5. Vælg en matrix som resultatmatrix. Det kan udmærket være samme matrix, som bruges til lagring af $\tilde{\mathbf{Z}}$.
6. Tryk på $\boxed{1/x}$. Herved beregnes $(\mathbf{Z})^{-1}$, som er identisk med $(\mathbf{Z}^{-1})$. Resultatet lagres i resultatmatricen hvis deskriptor lagres i $\mathbf{X}$.
7. Brug $\boxed{\text{f}} \boxed{\text{MATRIX}} 3$ til at transformere $(\tilde{\mathbf{Z}})^{-1}$ til $(\mathbf{Z}^{-1})^P$.
8. Hvis De ønsker den inverse på C -form, kan De bruge $\boxed{\text{g}} \boxed{\text{Cy},x}$.

De kan aflede de komplekse elementer i $\mathbf{Z}^{-1}$ ved at fremkalde elementerne i $\mathbf{Z}^P$ eller $\mathbf{Z}^C$ og kombinere disse som tidligere omtalt.

Eksempel: Beregn den inverse af den komplekse matrix $\mathbf{Z}$ (fra forrige eksempel).

$$\mathbf{A} = \mathbf{Z}^P = \begin{bmatrix} 4 & 7 \\ 1 & 3 \\ 3 & -2 \\ 5 & 8 \end{bmatrix}.$$

Indtastning

$\boxed{\text{RCL}} \boxed{\text{MATRIX}} \boxed{\text{A}}$

$\boxed{\text{f}} \boxed{\text{MATRIX}} 2$

Lyspanel

A **4** **2**

A **4** **4**

Deskriptoren for $\mathbf{A}$ lagres i $\mathbf{X}$.

Transformerer $\mathbf{Z}^P$ til $\tilde{\mathbf{Z}}$ og re-dimensionerer $\mathbf{A}$.

Indtastning

f RESULT B

1/x

f MATRIX 3

Lyspanel

A 4 4

b 4 4

b 4 2

B bruges som resultatmatrix.

Beregner $(\tilde{Z})^{-1} = \widetilde{(Z^{-1})}$ og lagrer resultatet i B.Transformerer $(\tilde{Z}^{-1})$ til $(Z^{-1})^P$.

Repræsentationen af Z^{-1} er lagret på partitioneret form i matrix B:

$$B = \left[\begin{array}{cc|cc} -0.0254 & 0.2420 & & \\ -0.0122 & -0.1017 & & \\ \hline -0.2829 & -0.0022 & & \\ 0.1691 & -0.1315 & & \end{array} \right] \left\{ \begin{array}{l} \text{Reel del} \\ \text{Imaginær del} \end{array} \right.$$

Multiplikation af komplekse matricer

Matrixproduktet af to komplekse matricer kan beregnes ved at udnytte sammenhængen $(YX)^P = \tilde{Y}X^P$.

YX , hvor Y og X er komplekse matricer, beregnes således:

1. Y 's og X 's elementer lagres i dataregistre – på C - eller P -form.
2. Deskriptoren for Y lagres i X .
3. Hvis matrix Y blev lagret på C -form, skal f $P_{Y,X}$ bruges til at transformere til Y^P .
4. Tryk på f MATRIX 2, så Y^P transformeres til $\tilde{Y}$.
5. Deskriptoren for X lagres i X .
6. Hvis matrix X blev lagret på C -form, skal f $P_{Y,X}$ bruges til at transformere til X^P .
7. Vælg en matrix som resultatmatrix. Det må ikke være en af de to matricer, som bruges til lagring af de andre to.
8. Tryk på $\times$. Herved beregnes $\tilde{Y}X^P = (YX)^P$. Resultatet lagres i resultatmatricen hvis deskriptor lagres i X .
9. Hvis De ønsker produktet på C -form, kan De bruge g $C_{Y,X}$.

Bemærk, at $\mathbf{X}^P$ ikke skal transformeres til $\tilde{\mathbf{X}}$.

De kan aflede de komplekse elementer i $\mathbf{YX}$ ved at fremkalde elementerne i $(\mathbf{YX})^P$ eller $(\mathbf{YX})^C$ og kombinere disse som tidligere omtalt.

Eksempel: Beregn produktet $\mathbf{ZZ}^{-1}$, hvor $\mathbf{Z}$ er den komplekse matrix, som blev brugt i foregående eksempel.

Da begge matrices elementer allerede er lagret ($\tilde{\mathbf{Z}}$ i **A** og $(\mathbf{Z}^{-1})^P$ i **B**) skal De ikke udføre trinene 1, 3, 4 og 6 i den generelle procedure.

Indtastning

RCL	MATRIX	A
RCL	MATRIX	B
f	RESULT	C
×		
f	USER	
RCL	C	

RCL	C
RCL	C
RCL	C
RCL	C
RCL	C
RCL	C
RCL	C
f	USER

Lyspanel

A	4	4	Udlæser A's deskriptor.
b	4	2	Udlæser B's deskriptor.
b	4	2	C bruges som resultatmatrix.
C	4	2	Beregner $\tilde{\mathbf{Z}}(\mathbf{Z}^{-1})^P = (\mathbf{ZZ}^{-1})^P$.
C	4	2	Skifter til USER-status.
C	1.1		Matrix C, række 1, søjle 1. (Udlæses medens sidste tast holdes nedtrykket.)
	1,0000		Værdien af $c_{1,1}$.
	-2,8500	-10	Værdien af $c_{1,2}$.
	-4,0000	-11	Værdien af $c_{2,1}$.
	1,0000		Værdien af $c_{2,2}$.
	1,0000	-11	Værdien af $c_{3,1}$.
	3,8000	-10	Værdien af $c_{3,2}$.
	1,0000	-11	Værdien af $c_{4,1}$.
	-1,0500	-10	Værdien af $c_{4,2}$.
	-1,0500	-10	Ophæver USER-status.

Vi skriver nu $\mathbf{C}$'s elementer:

$$\mathbf{C} = \begin{bmatrix} 1.0000 & -2.8500 \times 10^{-10} \\ -4.0000 \times 10^{-11} & 1.0000 \\ 1.0000 \times 10^{-11} & 3.8000 \times 10^{-10} \\ 1.0000 \times 10^{-11} & -1.0500 \times 10^{-10} \end{bmatrix} = (\mathbf{ZZ}^{-1})^P,$$

hvor den øverste halvdel af $\mathbf{C}$ er den reelle del af $\mathbf{ZZ}^{-1}$ og den nedre halvdel er den imaginære del. Vi får således, at $\mathbf{C}$ er

$$\mathbf{ZZ}^{-1} = \begin{bmatrix} 1.0000 & -2.8500 \times 10^{-10} \\ -4.0000 \times 10^{-11} & 1.0000 \end{bmatrix} \\ + i \begin{bmatrix} 1.0000 \times 10^{-11} & 3.8000 \times 10^{-11} \\ 1.0000 \times 10^{-11} & -1.0500 \times 10^{-10} \end{bmatrix}$$

Som forventet gælder:

$$\mathbf{ZZ}^{-1} \cong \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + i \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Løsning af den komplekse ligning $\mathbf{AX} = \mathbf{B}$

De kan løse den komplekse ligning $\mathbf{AX} = \mathbf{B}$ ved at finde $\mathbf{X} = \mathbf{A}^{-1}\mathbf{B}$. Dette gøres ved at beregne $\mathbf{X}^P = (\hat{\mathbf{A}})^{-1}\mathbf{B}^P$.

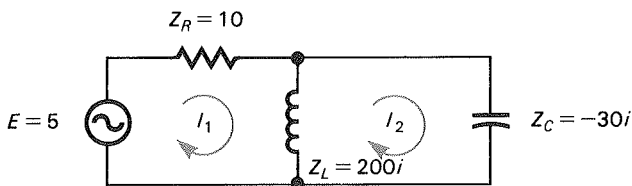
Ligningen $\mathbf{AX} = \mathbf{B}$, hvor $\mathbf{A}$, $\mathbf{X}$ og $\mathbf{B}$ er komplekse matricer, løses således:

1. $\mathbf{A}$'s og $\mathbf{B}$'s elementer lagres på C - eller P -form.
2. Fremkald deskriptoren for $\mathbf{B}$.
3. Hvis $\mathbf{B}$'s elementer blev lagret på C -form, skal De transformere $\mathbf{B}^C$ til $\mathbf{B}^P$ ved hjælp af `f` `Py,x`.
4. Fremkald deskriptoren for $\mathbf{A}$.
5. Hvis elementerne i $\mathbf{A}$ blev lagret på C -form, skal De bruge `f` `Py,x` til at transformere $\mathbf{A}^C$ til $\mathbf{A}^P$.
6. Tryk `f` `MATRIX` 2, så $\mathbf{A}^P$ transformeres til $\hat{\mathbf{A}}$.
7. Vælg en resultatmatrix – som ikke må være den samme som bruges til at repræsentere $\mathbf{A}$.
8. Tryk på `÷`; herved beregnes $\mathbf{X}^P$. Dens elementer lagres i resultatmatricen, hvis deskriptor bliver udlæst.
9. Hvis De ønsker at få løsningen på C -form, skal De trykke `g` `Cy,x`.

Bemærk at De ikke skal transformere $\mathbf{B}^P$ til $\tilde{\mathbf{B}}$.

De kan aflede de komplekse elementer af løsningen $\mathbf{X}$ ved at fremkalde elementerne i $\mathbf{X}^P$ eller $\mathbf{X}^C$ og kombinere dem som tidligere omtalt.

Eksempel: A.C. Poulsen, der læser elektronik, vil analysere det elektriske kredsløb, der vises nedenfor. Komponenternes impedans er givet på kompleks form. Bestem den komplekse repræsentation af strømmene I_1 og I_2 .



Systemet kan repræsenteres ved hjælp af de komplekse matrixligninger:

$$\begin{bmatrix} 10 + 200i & -200i \\ -200i & (200 - 30)i \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$$

eller

$$\mathbf{A}\mathbf{X} = \mathbf{B}.$$

På partitioneret form har vi

$$\mathbf{A} = \begin{bmatrix} 10 & 0 \\ 0 & 0 \\ \hline 200 & -200 \\ -200 & 170 \end{bmatrix} \quad \text{og} \quad \mathbf{B} = \begin{bmatrix} 5 \\ 0 \\ \hline 0 \\ 0 \end{bmatrix},$$

hvor nulelementerne svarer til reel- og imaginærdele med værdien 0.

Indtastning	Lyspanel
4 [ENTER] 2 [f] [DIM] [A]	2,0000
[f] [MATRIX] 1	2,0000
[f] [USER]	2,0000
10 [STO] [A]	10,0000
0 [STO] [A]	0,0000
[STO] [A]	0,0000
[STO] [A]	0,0000
200 [STO] [A]	200,0000
[CHS] [STO] [A]	- 200,0000
[STO] [A]	- 200,0000
170 [STO] [A]	170,0000
4 [ENTER] 1 [f] [DIM] [B]	1,0000
0 [STO] [MATRIX] [B]	0,0000
5 [ENTER] 1 [ENTER]	1,0000
[STO] [g] [B]	5,0000
[RCL] [MATRIX] [B]	b 4 1
[RCL] [MATRIX] [A]	A 4 2
[f] [MATRIX] 2	A 4 4
[f] [RESULT] [C]	A 4 4
[÷]	C 4 1
[g] [Cy,x]	C 2 2
[RCL] [C]	0,0372
[RCL] [C]	0,1311
[RCL] [C]	0,0437
[RCL] [C]	0,1543
[f] [USER]	0,1543
[f] [MATRIX] 0	0,1543

A har dimensionen 4×2 .

Gør klar til lagring af elementerne.

Skifter til USER-status.

Lagrer $a_{1,1}$.

Lagrer $a_{1,2}$.

Lagrer $a_{2,1}$.

Lagrer $a_{2,2}$.

Lagrer $a_{3,1}$.

Lagrer $a_{3,2}$.

Lagrer $a_{4,1}$.

Lagrer $a_{4,2}$.

B dimensioneres til 4×1 .

Matrix B nulstilles.

Elementet i række 1, søjle 1 skal være 5.

5 lagres i $b_{1,1}$. (Husk [g], når Y og X bruges til at lagre række- og søjlenumre.)

Deskriptoren for B.

Deskriptoren for A lagres i X, medens deskriptoren for B rykker op i Y.

A^P transformeres til $\tilde{A}$.

C bruges som resultatmatrix.

X^P beregnes og lagres i C.

X^P transformeres til X^C .

Fremkald af $c_{1,1}$.

Fremkald af $c_{1,2}$.

Fremkald af $c_{2,1}$.

Fremkald af $c_{2,2}$.

Ophæver USER-status.

Redimensionerer alle matrixer til 0×0 .

Strømmene, som repræsenteret ved den komplekse matrix $\mathbf{X}$, kan afledes af $\mathbf{C}$:

$$\mathbf{X} = \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} 0.0372 + 0.1311i \\ 0.0437 + 0.1543i \end{bmatrix}.$$

Løsning af matrixligningen i det foregående eksempel krævede 24 data-registre til lagring af matrixelementer – 16 til 4×4 matrixen $\mathbf{A}$ (som oprindelig blev lagret som en 4×2 matrix, der repræsenterede en 2×2 kompleks matrix) og fire til hver af matrixerne $\mathbf{B}$ og $\mathbf{C}$ (som hver repræsenterer en 2×1 kompleks matrix). (De kunne have klaret Dem med fire registre mindre, hvis De havde brugt $\mathbf{B}$ som resultatmatrix.) Bemærk, at da $\mathbf{X}$ og $\mathbf{B}$ ikke er begrænsede til at være vektorer, kunne $\mathbf{X}$ og $\mathbf{B}$ have krævet endnu mere lagerplads.

HP-15C har tilstrækkelig hukommelse til at løse, med ovenstående metode, den komplekse matrixligning $\mathbf{AX} = \mathbf{B}$, hvor $\mathbf{X}$ og $\mathbf{B}$ hver kan have indtil 6 søjler, hvis $\mathbf{A}$ er 2×2 , eller op til 2 søjler, hvis $\mathbf{A}$ er 3×3 .* (Det mulige antal søjler fordobles, hvis $\mathbf{B}$ bruges som resultatmatrix.) Hvis $\mathbf{X}$ og $\mathbf{B}$ har flere søjler, eller hvis $\mathbf{A}$ er 4×4 , kan De løse ligningen ved hjælp af metoden som beskrives nedenfor. Denne metode adskiller sig fra den foregående ved at den udnytter særskilt invertering og multiplikation til at spare registerplads.

1. $\mathbf{A}$'s elementer lagres på $\mathbf{C}$ - eller $\mathbf{P}$ -form.
2. Fremkald deskriptoren for $\mathbf{A}$.
3. Hvis $\mathbf{A}$'s elementer blev lagret på $\mathbf{C}$ -form, skal De transformere $\mathbf{A}^{\mathbf{C}}$ til $\mathbf{A}^{\mathbf{P}}$ ved hjælp af $\boxed{\text{f}} \boxed{\text{Py,x}}$.
4. Tryk $\boxed{\text{f}} \boxed{\text{MATRIX}} \boxed{2}$, så $\mathbf{A}^{\mathbf{P}}$ transformeres til $\tilde{\mathbf{A}}$.
5. Brug $\boxed{\text{STO}} \boxed{\text{RESULT}}$ til at vælge $\mathbf{A}$ som resultatmatrix.
6. Brug $\boxed{1/x}$ til at beregne $(\tilde{\mathbf{A}})^{-1}$.
7. Redimensionér $\mathbf{A}$, så antallet af rækker (som blev vist under udlæsningen af deskriptoren) bliver halveret.

* Hvis hele lageret er udlagt til matrixoperationer ($\boxed{\text{g}} \boxed{\text{MEM}} \boxed{1\ 64\ 0-0}$). Se tillæg C.

8. **B**'s elementer lagres på *C*- eller *P*-form.
9. Fremkald deskriptoren for **A**.
10. Fremkald deskriptoren for **B**.
11. Hvis elementerne i **B** blev lagret på *C*-form, skal De bruge $\boxed{f} \boxed{Py,x}$ til at transformere $\mathbf{B}^C$ til $\mathbf{B}^P$.
12. Tryk $\boxed{f} \boxed{\text{MATRIX}} \boxed{2}$, så $\mathbf{B}^P$ transformeres til $\tilde{\mathbf{B}}$.
13. Væg en resultatmatrix, som ikke falder sammen med **A** eller **B**.
14. Tryk på $\boxed{\times}$.
15. Transponér resultatmatricen med $\boxed{f} \boxed{\text{MATRIX}} \boxed{4}$.
16. Brug $\boxed{f} \boxed{\text{MATRIX}} \boxed{2}$.
17. Redimensionér resultatmatricen, så antallet af rækker halveres.
18. Brug $\boxed{\text{RCL}} \boxed{\text{RESULT}}$ til at fremkalde resultatmatricens deskriptor.
19. Brug $\boxed{f} \boxed{\text{MATRIX}} \boxed{4}$ til at beregne $\mathbf{X}^P$.
20. Hvis De ønsker at få løsningen på *C*-form, skal De trykke $\boxed{g} \boxed{Cy,x}$.

I *HP-15C Advanced Functions Handbook* kan De under *Solving a Large System of Complex Equations* finde et eksempel på brug af denne metode.

Forskellige operationer med matricer

Dataregisteroperationer med matricielementer

Hvis en bogstavtast, som specificerer en matrix, bliver brugt efter en af følgende funktionstaster, bliver operationen udført på det matricielement, hvis række og søjle specificeres i R_0 og R_1 – præcis som om De havde adresseret et dataregister på sædvanlig vis.

$\boxed{\text{STO}}^*$
 $\boxed{\text{STO}} \{ \boxed{+}, \boxed{-}, \boxed{\times} \text{ og } \boxed{\div} \}$
 $\boxed{\text{DSE}}$
 $\boxed{x \geq}$

$\boxed{\text{RCL}}^*$
 $\boxed{\text{RCL}} \{ \boxed{+}, \boxed{-}, \boxed{\times} \text{ og } \boxed{\div} \}$
 $\boxed{\text{ISG}}$

* I USER-status vil række- og søjlenumrene i R_0 og R_1 løbende blive justeret i henhold til matrixdimensionen.

Matrixdeskriptorer i indeksregistret

I forbindelse med visse anvendelser vil De få brug for at udføre en række matrixoperationer, hvor matricerne A til E bruges i rækkefølge. I sådanne tilfælde kan De anvende indeksregistret (R_1) til at lagre matrixdeskriptoren for den matrix, der skal arbejdes med.

Hvis indeksregistret indeholder en matrixdeskriptor gælder følgende:

- Brug af $\boxed{f} \boxed{(i)}$ efter enhver af de funktioner, der er anført ovenfor, udfører funktionen på elementet med rækkenummer i R_0 , søjlenummer i R_1 og matrixdeskriptor i R_1 .
- Brug af $\boxed{f} \boxed{(i)}$ efter $\boxed{STO} \boxed{g}$ eller $\boxed{RCL} \boxed{g}$ udfører operationen på elementet med rækkenummer i Y, søjlenummer i X og matrixdeskriptor i R_1 .
- Brug af $\boxed{f} \boxed{DIM} \boxed{I}$ dimensionerer matricen, der er specificeret i R_1 , efter række/søjlenumrene i Y og X.
- Brug af $\boxed{RCL} \boxed{DIM} \boxed{I}$ lagrer dimensionen af den matrix, som er specificeret i R_1 , i Y og X.
- Brug af $\boxed{GSB} \boxed{I}$ eller $\boxed{GTO} \boxed{I}$ har samme effekt som tryk på $\boxed{GSB}$ eller $\boxed{GTO}$ efterfulgt af den bogstavtast, som svarer til matricen, hvis deskriptor er lagret i R_1 . (Dette kan ikke kaldes en matrixoperation, da kun matricens bogstav – dens navn – bliver brugt.)

Sammenligningsoperationer med matrixdeskriptorer

De kan benytte fire betingelsesoperationer, $\boxed{x=0}$, $\boxed{TEST} \boxed{0} (x \neq 0)$, $\boxed{TEST} \boxed{5} (x = y)$ og $\boxed{TEST} \boxed{6} (x \neq y)$, på matricer, hvis deskriptorer er lagret i X og Y. Disse operationer kan bruges til at styre programforløb, således som beskrevet i afsnit 8.

Hvis en matrixdeskriptor er lagret i X, vil resultatet af $\boxed{x=0}$ være falsk og resultatet af $\boxed{TEST} \boxed{0}$ være sandt (uanset elementernes værdi).

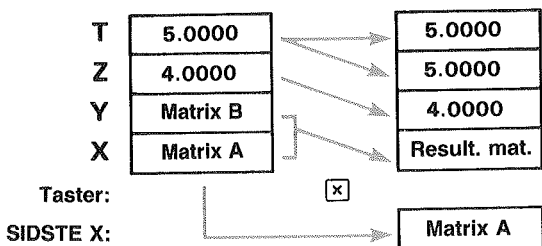
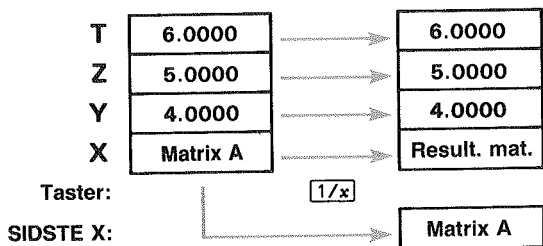
Hvis der er lagret matrixdeskriptorer i X og Y, når $\boxed{TEST} \boxed{5}$ eller $\boxed{TEST} \boxed{6}$ bliver udført, vil x og y blive betragtet som identiske, hvis deskriptorerne er identiske – ellers ikke. Dette gælder uanset om elementer i to forskellige deskriptorer er ens.

Sammenligningsoperationer på matricer udføres altså alene på deskriptorer og ikke på deres associerede elementer.

Stakoperationer ved matrixoperationer

Under matrixoperationer skifter stakregistrenes indhold på en måde, der meget ligner den, De allerede kender fra almindelige operationer.

I forbindelse med nogle beregninger bliver resultatet lagret i en resultatmatrix. Argumenterne – én eller to deskriptorer eller tal i X- eller Y-registrene – kombineres under operationen, og deskriptoren for resultatmatricen bliver lagret i X. (Argumentet fra X bliver som sædvanligt lagret i SIDSTE X.)

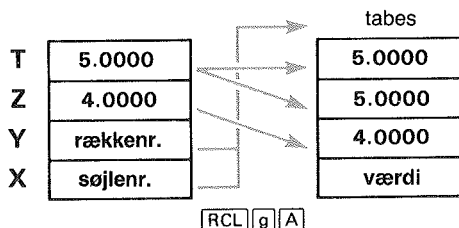
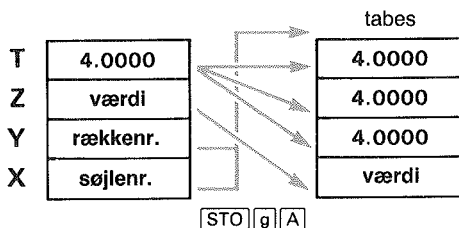


Adskillige af matrixfunktionerne virker alene på matricen, som er specificeret i X-registret og lagrer resultatet i samme matrix. I disse tilfælde bevæges stakken ikke (operationerne svarer aldeles til funktioner af én variabel) – om end lyspanelets indhold ændres, hvis det er nødvendigt at redimensionere.

Ved brug af **MATRIX** 7, **MATRIX** 8 og **MATRIX** 9 vil den oprindelige matrixdeskriptor blive lagret i SIDSTE X-registret, medens deskriptoren for normen eller (for **MATRIX** 9) determinanten placeres i X. Y, Z og T berøres ikke.

Når De fremkalder deskriptorer eller matrixelementer og stakløften er aktiv, vil deskriptorer og tal i stakken blive løftet – og T-registrets indhold tabes. (SIDSTE X-registret påvirkes derimod ikke.) Når De lagrer matrixdeskriptorer eller -elementer, påvirkes stak og SIDSTE X-register ikke.

I kontrast til ovenstående vil **STO** **g** og **RCL** **g** ikke påvirke SIDSTE X og virker i øvrigt som vist nedenfor.

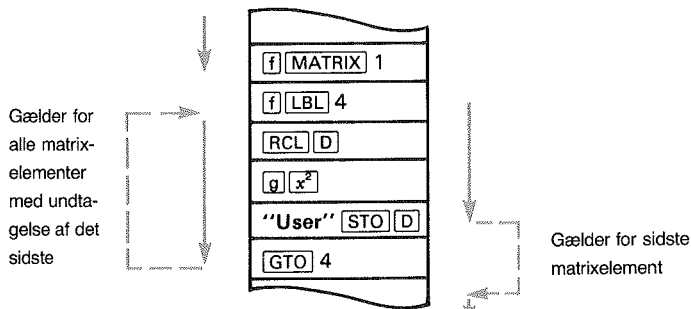


Brug af matrixoperationer i programmer

Hvis lommeregneren er i USER-status under programmering og De bruger **STO** eller **RCL** {**A** til **E**, **(i)**} til at fremkalde eller lagre et matrixelement, vil **u** blive udlæst i stedet for den sædvanlige bindestreg, som vises efter linienummeret. Når linien udføres under programkontrol, vil den virke som om USER-status var valgt. Det betyder, at række- og søjlenumrene i R_0 og R_1 automatisk forøges i overensstemmelse med den specificerede matrices dimension. På denne måde kan De bearbejde ele-

menterne i rækkefølge. (USER-indikatoren har ingen betydning under programudførelse.)

Dertil kommer en særlig finesse: Når R_0 og R_1 begge sættes til 1 efter at alle elementer er behandlet, bliver linien efter **RCL** eller **STO** sprunget over. De kan således programmere en løkke, som lagrer eller fremkalder de enkelte elementer i en matrix og derefter fortsætter programkørslen. F.eks. vil følgende program kvadrere alle elementer i matrix **D**:



MATRIX 7 (rækkenorm) og **MATRIX** 8 (Euklidisk norm) kan desuden virke som betingede forgreninger i et program. Hvis **X** indeholder en matrixdeskriptor, vil disse funktioner beregne matrixens norm og programkørslen fortsætter normalt. Men hvis **X** indeholder et tal, vil den følgende linie blive sprunget over. I begge tilfælde vil det originale indhold af **X**-registret blive lagret i **SIDSTE X**. På denne måde kan De teste, om **X**-registrets indhold er en deskriptor eller et rent tal.

Oversigt over matrixfunktioner

Indtastning(er)

Resultat

g **Cy,x**

Transformerer Z^P til Z^C .

CHS

Skifter fortegn på alle elementer i matricen, hvis deskriptor er lagret i **X**.

f **DIM** {**A** til **E**, **I**}

Dimensionerer den specificerede matrix.

f **MATRIX** 0

Dimensionerer alle matricer til 0×0 .

f **MATRIX** 1

Sætter række- og søjlenumrene (i R_0 og R_1) til 1.

Indtastning(er)	Resultat
$\boxed{f}$ $\boxed{\text{MATRIX}}$ 2	Transformerer $\mathbf{Z}^P$ til $\tilde{\mathbf{Z}}$.
$\boxed{f}$ $\boxed{\text{MATRIX}}$ 3	Transformerer $\tilde{\mathbf{Z}}$ til $\mathbf{Z}^P$.
$\boxed{f}$ $\boxed{\text{MATRIX}}$ 4	Beregner den transponerede af matricen i X.
$\boxed{f}$ $\boxed{\text{MATRIX}}$ 5	Ganger den transponerede af matricen, hvis deskriptor er lagret i Y, med matricen, hvis deskriptor er lagret i X. Lagrer resultatet i resultatmatricen.
$\boxed{f}$ $\boxed{\text{MATRIX}}$ 6	Beregner restmatricen.
$\boxed{f}$ $\boxed{\text{MATRIX}}$ 7	Beregner rækkenormen for matricen, hvis deskriptor er lagret i X.
$\boxed{f}$ $\boxed{\text{MATRIX}}$ 8	Beregner den Euklidiske norm af matricen, hvis deskriptor er lagret i X.
$\boxed{f}$ $\boxed{\text{MATRIX}}$ 9	Beregner determinanten for matricen, hvis deskriptor er lagret i X. Lagrer LU i resultatmatricen.
$\boxed{f}$ $\boxed{\text{Py},x}$	Transformerer $\mathbf{Z}^C$ til $\mathbf{Z}^P$.
$\boxed{\text{RCL}}$ $\boxed{\{ \text{A} \}}$ til $\boxed{\text{E}}, \boxed{(i)}$	Fremkalder element fra den specificerede matrix, idet R_0 og R_1 bruges til at angive række- og søjlenummer.
$\boxed{\text{RCL}}$ $\boxed{g}$ $\boxed{\{ \text{A} \}}$ til $\boxed{\text{E}}, \boxed{(i)}$	Fremkalder element fra den specificerede matrix, idet tallene i Y og X bruges til at angive række- og søjlenummer.
$\boxed{\text{RCL}}$ $\boxed{\text{DIM}}$ $\boxed{\{ \text{A} \}}$ til $\boxed{\text{E}}, \boxed{\text{I}}$	Lagrer dimensionen af den specificerede matrix i Y og X.
$\boxed{\text{RCL}}$ $\boxed{\text{MATRIX}}$ $\boxed{\{ \text{A} \}}$ til $\boxed{\text{E}}$	Udlæser deskriptoren for den specificerede matrix.
$\boxed{\text{RCL}}$ $\boxed{\text{RESULT}}$	Udlæser deskriptoren for resultatmatricen.
$\boxed{f}$ $\boxed{\text{RESULT}}$ $\boxed{\{ \text{A} \}}$ til $\boxed{\text{E}}$	Gør den specificerede matrix til resultatmatrix.
$\boxed{\text{STO}}$ $\boxed{\{ \text{A} \}}$ til $\boxed{\text{E}}, \boxed{(i)}$	Lagrer det udlæste tal som element i den specificerede matrix, idet R_0 og R_1 bruges til at angive række- og søjlenummer.

Indtastning(er)

STO **g** { **A** til **E**, **(i)** }

STO **MATRIX**
{ **A** til **E** }

STO **RESULT**

f **USER**

1/x

+, **-**

×

÷

Resultat

Lagrer tallet i **Z** i den specificerede matrix, idet tallene i **Y** og **X** bruges til at angive række- og søjlenummer.

Hvis en matrixdeskriptor udlæses, bliver matricen kopieret over i en specificeret matrix. Hvis der udlæses et tal, vil dette tal blive lagret i alle den specificerede matrices elementer.

Gør matricen, hvis deskriptor er lagret i **X**, til resultatmatrix.

Ændrer automatisk på indholdet af R_0 og R_1 , således at der gøres klar til næste element, hver gang **STO** eller **RCL** { **A** til **E**, **(i)** } bruges.

Beregner den inverse af den matrix, hvis deskriptor er lagret i **X**. Resultatet lagres i resultatmatricen.

Hvis der er lagret en matrixdeskriptor i såvel **Y** som **X**, vil elementerne i matrix **X** blive lagt til, henholdsvis trukket fra elementerne i **Y**. Hvis der kun er lagret en deskriptor i det ene stakregister, vil addition/subtraktion blive udført som en skalær operation på alle elementer. Resultatet lagres i resultatmatricen.

Hvis der er lagret en matrixdeskriptor i såvel **Y** som **X**, vil elementerne i matrix **X** blive lagt til, henholdsvis trukket fra elementerne i **Y**. Hvis der kun er lagret en deskriptor i det ene stakregister, vil addition/subtraktion blive udført som en skalær operation på alle elementer. Resultatet lagres i resultatmatricen.

Hvis der er lagret en matrixdeskriptor i såvel **Y** som **X**, vil den inverse af matrix **X** blive multipliceret med **Y**. Hvis der kun er lagret en deskriptor i **Y**, vil alle elementer i den specificerede matrix blive divideret med tallet i **X**. Hvis der kun er lagret en deskriptor i **X**, vil hvert element i X^{-1} blive multipliceret med skalaren i **Y**. Resultatet lagres i resultatmatricen.

Yderligere information

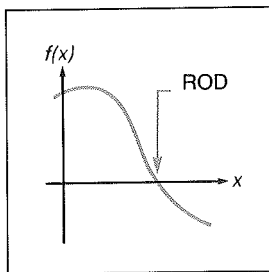
I *HP-15C Advanced Functions Handbook* kan De læse om flere og mere teknisk detaljerede aspekter af matrixfunktionerne. Det drejer sig f.eks. om mindste kvadraters metode, løsning af ikke-lineære ligninger, dårligt konditionerede eller singulære matricer, nøjagtighedsovervejelser, iterativ finpudsning og den identiske matrix.

Løsning af ligninger

I mange anvendelser vil De få brug for at løse ligninger på formen:

$$f(x) = 0.*$$

Det betyder, at De skal finde de værdier af x , som tilfredsstiller ligningen. Enhver sådan værdi af x kaldes en *rod* i $f(x)$ eller et *nulpunkt* for $f(x)$. Disse rødder (eller nulpunkter) kaldes *reelle rødder* (eller reelle nulpunkter). I mange tilfælde kan en rod findes analytisk ved hjælp af passende algebraiske metoder. Men i mange andre tilfælde er dette ikke muligt. De kan anvende numeriske metoder til at estimere rødderne, når analytiske metoder ikke er anvendelige. Når De bruger **SOLVE** på HP-15C, anvender De en meget avanceret numerisk teknik, som giver Dem en sikker og hurtig metode til bestemmelse af *reelle rødder* i et bredt spektrum af ligninger.**



Brug af **SOLVE**

Under beregning af rødder vil **SOLVE** få brug for et underprogram, *som De skal skrive*. Programmet skal beregne $f(x)$ for givne værdier.

* Det er faktisk således, at *enhver* funktion af én variabel kan udtrykkes på denne form. F.eks. er $f(x) = a$ ækvivalent med $f(x) - a = 0$, ligesom $f(x) = g(x)$ er ækvivalent med $f(x) - g(x) = 0$.

** **SOLVE** funktionen kan ikke arbejde med imaginærstakken. Se *HP-15C Advanced Functions Handbook* for en diskussion af komplekse rødder.

De grundlæggende regler for brugen af **SOLVE** er:

1. I programmeringsstatus skal De indtaste en tastsekvens, som kan beregne funktionen $f(x)$, som skal være sat til 0. Dette underprogram skal starte med en etiket **f** **LBL** og skal slutte med værdien af $f(x)$ i X.

I normalstatus:

2. Indtast to startværdier i X og Y. Startværdierne bruges alene til at angive et mål for det interval, hvor HP-15C skal søge rødder. Hvis intervallet er tilstrækkelig snævert og der faktisk findes rødder inden for intervallet, vil beregningstiden blive ganske kort.
3. Tryk på **SOLVE** efterfulgt af etiketnavnet for underprogrammet. Maskinen søger nu at finde en løsning og vil udlæse resultatet af sine beregninger. Hvis funktionen har mere end ét nulpunkt, vil programmet stoppe, så snart ét nulpunkt er fundet – uden indikation af, at der måske findes flere nulpunkter. De kan finde eventuelle andre nulpunkter ved at indtaste nye startværdier og igen trykke på **SOLVE**.

Umiddelbart før **SOLVE** adresserer underprogrammet bliver den aktuelle værdi af x lagret i X, Y, Z og T. Denne værdi bliver derefter brugt som variabel under beregningen af $f(x)$. Da hele stakken er fyldt op med x , vil tallet hele tiden være til rådighed for underprogrammet. (Brugen af denne teknik er beskrevet på side 41).

Eksempel: Brug **SOLVE** til at finde de værdier af x , som tilfredsstiller ligningen

$$f(x) = x^2 - 3x - 10 = 0.$$

Brug Horner's metode (se side 78) til at omskrive udtrykket til det mere beregningsvenlige:

$$f(x) = (x - 3)x - 10.$$

I programmeringsstatus skal De indtaste følgende underprogram til beregning af $f(x)$.

Indtastning

g **P/R**
f **CLEAR** **PRGM**

Lyspanel

000 –
000 –

Programmeringsstatus.
 Programhukommelsen slettes.

Indtastning

f LBL 0

3

-

x

1

0

=

g RTN

Lyspanel

001 - 42.21. 0

002 - 3

003 - 30

004 - 20

005 - 1

006 - 0

007 - 30

008 - 43 32

Start med LBL. Underprogrammet forudsætter, at x er lagret i alle stakregistre.

Beregner $x - 3$.Beregner $(x - 3)x$.Beregner $(x - 3)x - 10$.

Indtast to startværdier, når De har skiftet tilbage til normalstatus. Brug 0 og 10 for at angive interesse for positive rødder.

Indtastning

g P/R

0 ENTER

10

Lyspanel*

0,0000

10

Normalstatus.

} Startværdier.

Nu kan De finde den ønskede rod ved at trykke f SOLVE 0. Når De gør dette, vil lommeregneren ikke udlæse svaret med det samme. HP-15C anvender en iterativ** metode til at finde en tilnærmelse til roden. Algoritmen analyserer funktionen ved at beregne den for forskellige værdier af x - ofte 10 gange eller flere. Den gør det ved at gentage underprogrammet. Derfor vil det normalt tage fra 30 sekunder til 2 minutter for HP-15C at finde en rod; men somme tider tager det endnu længere tid.

Tryk på f SOLVE 0 og læn Dem tilbage medens HP-15C demonstrerer en af sine allerstærkeste funktioner. I lyspanelet blinker **running** medens beregningen står på.

* Brug f FIX 4 for at opnå den viste udlæsning. Valget af udlæsningsformat har ingen betydning for SOLVE.

** En *algoritme* er en beskrivelse, der trin for trin viser, hvorledes en bestemt proces skal udføres. En *iterativ* algoritme indeholder et afsnit som gentages et antal gange under processen. Typisk anvendes resultatet af en beregning som ny startværdi for iterationen indtil et bestemt kriterium er opfyldt.

Indtastning

f SOLVE 0

Lyspanel

5,0000

Den ønskede rod.

Når rutinen har fundet en rod, kan De kontrollere at det udlæste tal faktisk er en rod i $f(x)$ ved at inspicere stakken. De har set, at lyspanel-registret (X) indeholder roden. Y indeholder det foregående estimat for roden (den foregående tilnærmelse), som normalt vil være meget nær den udlæste tilnærmelse. I Z har De funktionsværdien, således som den beregnes med den udlæste tilnærmelse (roden).

Indtastning

R↓

R↓

Lyspanel

5,0000

0,0000

Det foregående rodestimat.

Funktionsværdien for

 $x = 5,0000$.

Kvadratiske ligninger, som den De netop har fundet en løsning til, kan have to rødder. Hvis De nu specificerer to nye startværdier (som skal dække et andet interval) kan De undersøge, om der findes en anden rod. Brug 0 og -10 til at lede efter en negativ rod.

Indtastning

0 ENTER

10 CHS

f SOLVE 0

R↓

R↓

Lyspanel

0,0000

- 10

- 2,0000

- 2,0000

0,0000

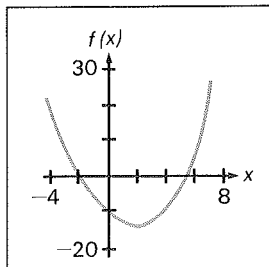
} Startværdier

Den anden rod.

Foregående tilnærmelse.

Værdien af $f(x)$ for $x =$ $- 2,0000$.

De har nu fundet to rødder i $f(x)$. Bemærk at denne ligning udmærket kunne være løst *analytisk* – og at De ville have opnået samme resultat.

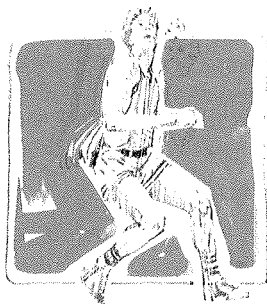
Graf for $f(x)$

Men De får først for alvor en følelse af styrken bag **SOLVE**, når De løser en ligning, som ikke kan løses analytisk (hvor den ubekendte simpelt hen ikke kan isoleres på den ene side af lighedstegnet).

Eksempel: Boldmester Castholm kan kaste en bold, så den har en lodret start-hastighed på 50 meter/sekund. Hvis kaste-højden er givet ved

$$h = 5000(1 - e^{-t/20}) - 200t,$$

hvor lang tid tager det da bolden at nå jorden? I formlerne er h højden og t tiden i sekunder.



Løsning: Den søgte løsning findes for den positive værdi af t , hvor $h = 0$. Brug følgende underprogram til at bestemme højden.

Indtastning

g **P/R**
f **LBL** **A**

2

0

÷
CHS

e^x
CHS

1

+

5

0

0

0

Lyspanel

000 -

001 - 42.21.11

002 - 2

003 - 0

004 - 10

005 - 16

006 - 12

007 - 16

008 - 1

009 - 40

010 - 5

011 - 0

012 - 0

013 - 0

Start med en etiket.

Underprogrammet forudsætter, at t er lagret i X og Y.

$-t/20$.

$-e^{-t/20}$.

$1 - e^{-t/20}$.

Indtastning

$\times$
 $x \approx y$
 2
 0
 0
 $\times$
 $-$
 g RTN

Lyspanel

014 – 20
 015 – 34
 016 – 2
 017 – 0
 018 – 0
 019 – 20
 020 – 30
 021 – 43 32

$5000(1 - e^{-t/20})$.
 Lægger en ny t -værdi i X.
 $200t$.
 $5000(1 - e^{-t/20}) - 200t$.

Skift til normalstatus, indtast to startværdier for tiden (f.eks. 5 og 6 sekunder) og udfør **SOLVE**.

Indtastning

g P/R
 5 ENTER
 6
 f SOLVE A

Lyspanel

5,0000
 6
 9,2843

Normalstatus.
 } Startværdier
 Tiden. (Bemærk, at roden ligger *uden* for startintervallet.)

Kontrollér roden ved at inspicere Y og Z.

Indtastning

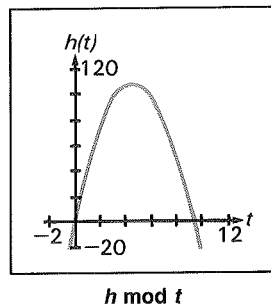
R
 R

Lyspanel

9,2843
 0,0000

Et tidligere estimat.
 h ved tiden $t = 9,2843$ sek.

Castholms bold rammer jorden 9,2843 sekunder efter kast. I sandhed et forbløffende kast.



Forgæves søgning efter nulpunkter

De har nu set, hvorledes **SOLVE** estimerer og udlæser rødder i en funktion på formen $f(x) = 0$. Men det er jo muligt, at udtrykket ikke har nogen reelle rødder (dvs. at der ikke findes et reelt x , som opfylder identiteten). I sådanne tilfælde vil HP-15C udlæse **Error 8** som tegn på, at ligningen ikke har løsninger.

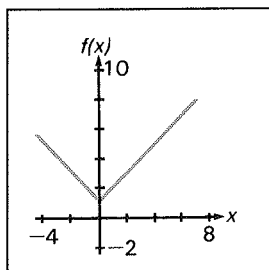
Eksempel: Betragt ligningen

$$|x| = -1$$

som ikke har nogen løsninger, da den numeriske værdi af et tal altid er positiv. Skriv funktionen på HP-15C's normalform

$$|x| + 1 = 0$$

og brug **SOLVE** til at forsøge at finde en reel løsning.



Graf for $f(x) = |x| + 1$

Indtastning

g P/R
f LBL 1
g ABS
1
+
g RTN

Lyspanel

000 -
001 - 42.21. 1
002 - 43 16
003 - 1
004 - 40
005 - 43 32

Programmeringsstatus.

Da funktionen har minimum for x nær 0, bør De vælge startværdier i nærheden af 0 - f.eks. -1 og 1. Prøv dernæst at finde en rod.

Indtastning

g P/R
1 ENTER
1 CHS
f SOLVE 1
←

Lyspanel

1,0000
-1
Error 8
0,0000

Normalstatus.

} Startværdier

Fejlmeddelelsen viser, at der ikke blev fundet en rod.
Fejltilstanden ophæves.

Som De kan se, stoppede HP-15C sin søgen efter en rod, da den besluttede, at der faktisk ikke findes nogen rod – i det mindste ikke i nærheden af det specificerede interval. **Error 8**-meddelelsen er ikke et tegn på, at der er udført en ”ugyldig” operation – kun, at **SOLVE** ikke kunne finde en rod i nærheden af det interval, der blev specificeret i forbindelse med løsningsprocessen.

I følgende tilfælde vil HP-15C opgive at finde en rod og udlæser en fejlmeddelelse:

- Gentagne iterationer har givet samme funktionsværdi, som er forskellig fra nul. **Error 8** bliver udlæst.
- Funktionsberegninger for adskillige værdier af x indikerer, at funktionsværdiens minimum *størrelsesordensmæssigt* er forskelligt fra 0. **Error 8** bliver udlæst.
- Der udføres en ugyldig operation under beregning af funktionen (f.eks. division med 0). **Error 0** bliver udlæst.

I de tilfælde, hvor funktionsværdien er konstant, kan rutinen ikke se nogen bevægelse mod nul. Denne situation kan indtræde, selv om funktionen faktisk ikke er konstant, hvis de første 10 betydende cifre er konstante. Det kan ske for en funktion med en vandret asymptote. Eller for en funktion, der er konstant over et interval, som er relativt bredt i forhold til det startinterval, der er specificeret.

I de tilfælde, hvor funktionsværdien stedse er forskellig fra nul, har rutinen beregnet funktionsværdier for værdier af x , der hele tiden har givet numerisk mindre og mindre funktionsværdier – uden på noget tidspunkt at nå nul eller skifte fortegn.

Den sidste mulighed vil normalt snarere antyde en utilstrækkelighed i det underprogram, som beregner funktionen, end en begrænsning i ligningsløseren. Ugyldige operationer kan i nogle tilfælde undgås ved at specificere et startinterval, som ikke indeholder et interval, hvor funktionen går mod uendelig eller ikke er defineret. Men **SOLVE** er ret aggressiv og går meget vidt omkring i sin søgen efter brugbare funktionsværdier. Det er derfor en god idé at lade underprogrammet teste argumenter for fejlskabende værdier og eventuelt justere dem (De kan f.eks. bruge **ABS** foran $\sqrt{x}$). På samme måde kan en skalering af meget store variable i visse tilfælde være til hjælp.

SOLVE's succes afhænger primært af den undersøgte funktions natur samt af de valgte startværdier. Den blotte eksistens af en rod sikrer ikke, at en tilfældig anvendelse af **SOLVE** vil finde roden. Hvis f.eks. funktionen $f(x)$ har en vandret asymptote, som er forskellig fra nul eller et lokalt minimum, kan man kun forvente at **SOLVE** finder en rod, hvis startværdierne dækker et andet interval end sådanne uproduktive områder.

Startværdier

Når De bruger **SOLVE** til at finde en funktions rødder, vil de to startværdier bestemme de værdier, den variable x antager, når rutinen indleder sin søgen efter en rod. I almindelighed vokser sandsynligheden for at De vil finde en ønsket rod med Deres forståelse af og viden om den funktion De analyserer. Realistiske og intelligente startværdier forkorter søgetiden og forøger chancen for at finde en rod.

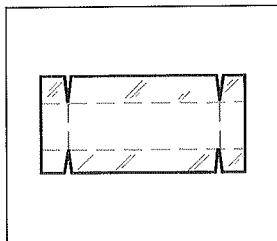
Startværdierne kan vælges på forskellig vis:

Hvis løsninger kun giver mening inden for et vist interval, vil det være rimeligt at anvende værdier inden for dette intervals grænser som startværdier. I mange tilfælde har en ligning, som beskriver et fysisk fænomen, en række løsninger, som ikke har nogen fysisk mening – f.eks. en negativ tidsværdi. Dette sker normalt fordi den funktion, der analyseres, netop kun er anvendelig for et begrænset interval. De bør altid mærke Dem sådanne begrænsninger og udnytte dem ved Deres valg af startværdier.

Hvis De har viden om funktionens adfærd, kan De i nogle tilfælde angive startværdier, som ligger meget nær en rod. På denne måde vil De ligeledes kunne undgå at specificere de mere besværlige intervaller, hvor funktionen antager et lokalt minimum eller er asymptotisk. Så selv om **SOLVE** i næsten alle tilfælde vil kunne finde en eksisterende rod, er den sædvanlige funktionelle analyse stadig af betydning.

Eksempel: En åben kasse med et volumen på $7,5 \text{ dm}^3$ skal laves af et stykke rektangulært metal med målene $4 \times 8 \text{ dm}$. Hvorledes skal metallet foldes? (En høj kasse foretrækkes frem for en lav.)

Løsning: De skal finde den højde (dvs. det stykke der foldes op som en af de fire sider), der giver det ønskede volumen.



Det synes rimeligt, at enten en høj og snæver kasse eller en bred og flad kan laves af det givne metalstykke. Og da den høje foretrækkes er det rimeligst at vælge store startværdier for højden. Højder over 2 dm er ikke fysisk mulige (da metalstykket kun er 4 dm bredt), så vi vil vælge startværdierne 1 og 2 dm.

Find den ønskede højde:

Indtastning**Lyspanel**

[g] [P/R]

1 [ENTER]

2

[f] [SOLVE] 3

[R↓]

[R↓]

1,0000**2****1,5000****1,5000****0,0000**

Normalstatus.

Startværdier

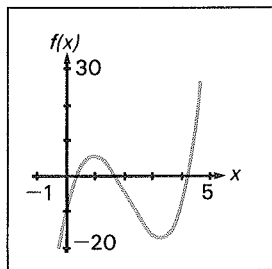
Den ønskede højde.

Foregående estimat.

Funktionsværdien.

Ved at sætte højden til 1,5 dm er der specificeret en kasse med målene $5,0 \times 1,0 \times 1,5$ dm.

Hvis De ignorerer den øvre grænse for højden og bruger 3 og 4 som startværdier, vil De få en højde på 4,2026 dm – en matematisk acceptabel rod som fra et fysisk synspunkt er meningsløs. Hvis De bruger mindre startværdier, som f.eks. 0 og 1, vil De få roden 0,2974 – den lave kasse.



Graf for $f(x)$

Som en hjælp under analysen af funktionens adfærd i forskellige intervaller kan De med fordel bruge det underprogram, De har lagret til brug for [SOLVE]. De skal blot fylde stakken med en værdi for x . Når De trykker [f] bogstavtast eller [GSB] etiket, bliver funktionsværdien beregnet for x .

De værdier, der således beregnes, kan De plotte i et sædvanligt koordinatsystem, så De får en skitse af funktionens graf. Denne indledende procedure er specielt værdifuld ved arbejde med funktioner, hvis adfærd De ikke kender. En uskyldigt udseende funktion kan være uhyre komplicerede og ekstreme variationer i netop det interval, hvor der findes en rod. En rod i nærheden af et variationsområde kan være uhyre vanskelig at finde, hvis ikke startværdierne ligger nær en rod.

Hvis De ikke kan anskueliggøre funktionen grafisk og ikke kan forestille Dem dens adfærd nær en rod, kan De forsøge Dem med mere eller mindre tilfældigt valgte værdier af x . Hvis De er heldig, vil De finde et mønster eller en tendens, som fører til et profitabelt startinterval. Hvor meget De skal hjælpe heldet, afhænger af funktionen.

- Hvis De specificerer to moderat store positive eller negative tal og funktionen ikke har en vandret asymptote, vil rutinen søge den numerisk største rod (medmindre funktionen oscillerer som f.eks. de trigonometriske funktioner).
- Hvis De allerede har fundet en rod, kan De lede efter andre rødder ved at specificere startværdier, som ligger relativt langt fra den eller de kendte nulpunkter.
- Mange funktioner udviser en speciel adfærd når deres argument går mod nul. De kan undersøge for hvilke x dele af funktionen går mod nul og bruge sådanne værdier eller nærtliggende værdier som startværdier.

Selv om De normalt specificerer to forskellige startværdier kan De udmærket anvende samme tal i X og Y. I sådanne tilfælde danner **SOLVE** selv et nyt estimat, systemestimatet. Hvis startværdien er forskellig fra nul, vil systemestimatet afvige fra startværdien med 1 på det syvende betydende ciffer. Hvis startværdien er 0, vil systemestimatet være 1×10^{-7} . Herefter fortsætter **SOLVE** på sædvanlig vis.

Brug af **SOLVE** i programmer

De kan bruge **SOLVE** som instruktion i et program. De skal blot sørge for, at programmet lagrer startværdier i X og Y inden **SOLVE** bruges. **SOLVE**-rutinen slutter med en x -værdi i X og den tilhørende funktionsværdi i Z. Hvis x -værdien er en rod, vil programpilen fortsætte til næste programlinje. Hvis x ikke er en rod, bliver linien efter **SOLVE** sprunget over. (Se også afsnittet om fortolkning af resultater side 223, hvor rod-begrebet omtales nærmere.) HP-15C kan altså selv afgøre, om der er fundet en rod eller ej. Testen kan udføres efter det sædvanlige Hop-hvis-JA-princip. Hvis **SOLVE** ikke finder en rod, kan maskinen således vælge et andet startinterval eller udføre en anden proces i forsøget på at finde en rod.

Når De anvender `SOLVE` i et program, bliver ét af de syv returadresse-registre brugt af rutinen. Samtidig afsluttes det underprogram, som kaldes af `SOLVE`, af en `RTN`-funktion. Det betyder, at der højst kan arbejdes med fem niveauer af underprogramkald, når `SOLVE` er aktiv under programkørsel. Hvis derimod `SOLVE` aktiveres fra tastaturet, bliver den første returadresse ikke brugt. Når `SOLVE` kaldes fra tastaturet, kan der således arbejdes med seks niveauer. Husk at når alle syv registre i returadressestakken er brugt vil forsøg på at kalde et underprogram resultere i **Error 5**. (Se side 104.)

Begrænsninger i brugen af `SOLVE`

Der er kun en begrænsning for brugen af `SOLVE`: funktionen kan ikke anvendes rekursivt – dvs. at `SOLVE` ikke kan bruges under udførelse af et underprogram, som er kaldt af `SOLVE`. Hvis De forsøger at bruge `SOLVE` rekursivt, bliver **Error 7** udlæst. Derimod er der intet i vejen for at bruge `SOLVE` sammen med `↵`.

Krav til lagerplads

`SOLVE` stiller krav om fem dataregistre. (I tillæg C forklares, hvorledes disse registre automatisk tages fra hukommelsen.) Hvis der ikke er fem tomme dataregistre til rådighed vil `SOLVE` ikke køre, og **Error 10** bliver udlæst.

En rutine, som bruger såvel `SOLVE` som `↵`, stiller krav om 23 ledige dataregistre.

Reference til yderligere information

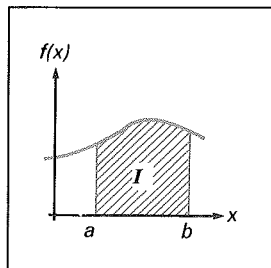
I tillæg D, Detaljeret gennemgang af `SOLVE`, gennemgås flere detaljer og anvendelser af `SOLVE` – blandt andet:

- Hvorledes `SOLVE` virker.
- Rodens nøjagtighed.
- Fortolkning af resultater.
- Søgning efter flere rødder.
- Begrænsning af søgetiden.

Numerisk integration

Mange opgaver inden for matematik, naturvidenskab og teknik kræver, at man skal beregne værdien af et bestemt integral. Hvis $f(x)$ er funktionen og a til b er integralets grænser, kan det bestemte integral skrives således:

$$I = \int_a^b f(x) dx.$$



Størrelsen I kan geometrisk fortolkes som arealet under kurven $f(x)$, dvs. det område, der er begrænset af linierne $x = a$, $x = b$, $y = 0$ og funktionen $f(x)$.*

Når et integral er vanskeligt eller umuligt at bestemme analytisk kan man tilnærme en løsning med numeriske metoder. Dette gøres sædvanligvis med ret store og komplicerede datamatprogrammer. Men det kan også gøres med HP-15C's integrationstast, $\int$.**

Brug af $\int$

Basisreglerne for brug af $\int$ er:

1. I programmeringsstatus skal der indtastes et underprogram, som beregner funktionen $f(x)$. Underprogrammet skal indledes med en etiket (f LBL etiket) og skal lagre x i lyspanelregistret, når det slutter.

* Forudsat at $f(x)$ er ikke-negativ over hele integrationsintervallet.

** $\int$ arbejder ikke med den komplekse stak. Se *HP-15C Advanced Functions Handbook* for information om brug af $\int$ i kompleksstatus.

I normalstatus:

2. Indtast den nedre grænse for det bestemte integral, a , og tryk på **ENTER** så a lagres i Y.
3. Indtast den øvre grænse, b , som således er lagret i X.
4. Tryk **f** **f** efterfulgt af etiketten for det underprogram, der beregner $f(x)$.

Eksempel: Visse problemer i fysikken kræver brug af *Bessels funktioner*. Bessels funktion af nulte orden kan udtrykkes således:

$$J_0(x) = \frac{1}{\pi} \int_0^\pi \cos(x \sin \theta) d\theta.$$

Find

$$J_0(1) = \frac{1}{\pi} \int_0^\pi \cos(\sin \theta) d\theta.$$

Indtastning

g **P/R**
f **CLEAR** **PRGM**
f **LBL** 0

SIN
COS
g **RTN**

Lyspanel

000 –
 000 –
 001 – 42.21. 0
 002 – 23
 003 – 24
 004 – 43 32

Programmeringsstatus.
 Programhukommelsen slettes.
 Underprogrammet indledes med **LBL**. Underprogrammet forudsætter, at θ er lagret i X.
 Beregning af $\sin \theta$.
 Beregning af $\cos(\sin \theta)$.

Skift nu til normalstatus og indtast først den nedre integrationsgrænse, tryk på **ENTER** og indtast dernæst den øvre grænse. Når De trykker **f** **fx** etiket, vil HP-15C starte beregningen af det bestemte integral. I dette specielle tilfælde skal De huske at vælge radian som vinkelmål inden De starter integrationen.

Indtastning

g **P/R**
0 **ENTER**
g **π**
g **RAD**

Lyspanel

0,0000
3,1416
3,1416

Normalstatus.
 Nedre grænse, 0.
 Øvre grænse, π .
 Radian vælges som vinkel-
 mål.

Nu er De klar til at trykke **f** **fx** 0, så integrationen kan begynde. De vil bemærke, at **fx**, ligesom **SOLVE**, ikke giver resultatet med det samme, som det ellers er tilfældet med HP-15C's funktioner. Integralet bestemmes ved hjælp af en uhyre avanceret iterativ metode. I store træk kan man sige, at rutinen bestemmer $f(x)$ for en lang række værdier af x i integrationsintervallet. Hver gang skal underprogrammet til bestemmelse af $f(x)$ udføres. Når maskinen skal udføre et sådant underprogram et stort antal gange må De ikke forvente et øjeblikkeligt svar. De fleste integraler beregnes i løbet af 30 sekunder til 2 minutter; men nogle integraler tager endnu længere tid. Vi vil senere fortælle, hvorledes De kan reducere beregningstiden noget. Men lige nu skal De blot trykke **f** **fx** 0 og tage en pause – eller læse videre – medens HP-15C bestemmer $J_0(1)$.

Indtastning

f **fx** 0

Lyspanel

2,4040

$$= \int_0^{\pi} \cos(\sin \theta) d\theta.$$

Hvis integralet indeholder en konstant faktor skal De huske at multiplicere resultatet med denne faktor. I vort tilfælde er den reciprokke værdi af π konstant, så vi udfører følgende division:

Indtastning

g **π**
 $\div$

Lyspanel

3,1416
0,7652

$$J_0(1).$$

Inden $\boxed{f}$ kalder underprogrammet til beregning af $f(x)$ vil x automatisk blive lagret i X, Y, Z og T – akkurat som ved $\boxed{\text{SOLVE}}$. Da alle stakregistre indeholder x kan underprogrammet regne med dette tal uden at skulle hente det fra et dataregister. Underprogrammerne i de næste to eksempler udnytter dette faktum. (En sådan teknik til bestemmelse af polynomier er gennemgået side 78.)

Bemærk: Da lommeregneren lagrer x i alle stakregistre, vil ethvert tidligere stakindhold blive overskrevet af x . Hvis De skal bruge tal fra stakken efter beregning af $f(x)$, må De derfor lagre sådanne mellemresultater i passende dataregistre.

I nogle tilfælde har De ikke brug for at bestemme integralet, I , men vil blot have funktionen $f(x)$ beregnet for en given værdi af x . I sådanne tilfælde skal De blot fylde stakken med x – ved at indtaste x og trykke $\boxed{\text{ENTER}}$ $\boxed{\text{ENTER}}$ $\boxed{\text{ENTER}}$ – inden De aktiverer underprogrammet.

Eksempel: Bessels funktion af første orden kan udtrykkes således:

$$J_1(x) = \frac{1}{\pi} \int_0^\pi \cos(\theta - x \sin \theta) d\theta.$$

Find $J_1(1) = \frac{1}{\pi} \int_0^\pi \cos(\theta - \sin \theta) d\theta.$

Indtast følgende underprogram, som bestemmer funktionen $f(\theta) = \cos(\theta - \sin \theta)$.

Indtastning

$\boxed{g}$ $\boxed{\text{P/R}}$
 $\boxed{f}$ $\boxed{\text{LBL}}$ 1

Lyspanel

000 –
 001 – 42.21. 1

Programmeringsstatus.
 Starter underprogrammet
 med etiket 1.

Indtastning

SIN

-

COS

g RTN

Lyspanel

002 - 23

003 - 30

004 - 24

005 - 43 32

 Beregning af $\sin \theta$.

 Da θ allerede er lagret i Y, når $\boxed{f} \boxed{\int}$ startes, vil $\boxed{-}$ beregne $(\theta - \sin \theta)$.

 $\cos(\theta - \sin \theta)$.

Skift til normalstatus og indtast integrationsgrænserne i X og Y. Sørg for at radian er valgt som vinkelmål og tryk $\boxed{f} \boxed{\int}$ 1, hvorefter integrationen startes. Gang resultatet med $1/\pi$, så $J_1(1)$ bliver beregnet.

Indtastning

g P/R

0 ENTER

 g π

g RAD

 $\boxed{f} \boxed{\int}$ 1

 g π $\div$
Lyspanel

0,0000

3,1416

3,1416

1,3825

0,4401

Normalstatus.

Nedre grænse lagres i Y.

Øvre grænse lagres i X.

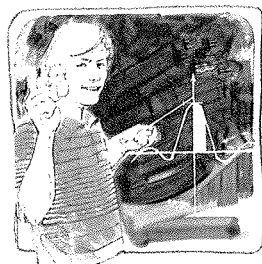
Radian som vinkelmål.

$$= \int_0^{\pi} \cos(\theta - \sin \theta) d\theta.$$

 $J_1(1)$.

Eksempel: Visse problemer inden for kommunikationsteori (f.eks. pulstransmission gennem ideale netværk) forudsætter at et særligt integral, *sinus*-integralet, beregnes:

$$Si(t) = \int_0^t \frac{\sin x}{x} dx.$$



Find $Si(2)$

Indtast følgende underprogram til beregning af funktionen

$$f(x) = (\sin x)/x.*$$

Indtastning

g **P/R**

f **LBL** .2

SIN

x $\rightrightarrows$ y

÷

g **RTN**

Lyspanel

000 –

001 – 42.21. ,2

002 – 23

003 – 34

004 – 10

005 – 43 32

Programmeringsstatus.

Underprogrammet indledes med **LBL** ,2.

Beregning af $\sin x$.

Da værdien af x er lagret i Y før **f** startes, vil **x $\rightrightarrows$ y** på dette punkt lagre x i X og $\sin x$ i Y.

$\sin x$ divideres med x .

Indtast nu integralets grænser i X og Y. Vælg radian som vinkelmaal og tryk **f** **f** ,2.

Indtastning

0 **ENTER**

2

g **RAD**

f **f** .2

Lyspanel

0,0000

2

2,0000

1,6054

Nedre grænse.

Øvre grænse.

Radian som vinkelmaal.

$Si(2)$.

* Hvis lommeregneren sættes til at beregne $f(x) = (\sin x)/x$ for $x = 0$, der er integralets nedre grænse, vil rutinen blive afbrudt og fejlmeddelelsen **Error 0** blive udlæst (som tegn på et forsøg på at dividere med 0). Men **f** vil normalt *ikke* beregne funktionsværdierne i det bestemte integrals grænsepunkter, så HP-15C kan faktisk bestemme et integral selv om det er ubestemt i en eller begge grænser. Hvis integrationsintervallet er ekstremt kort eller hvis antallet af beregnede funktionsværdier er ekstremt stort, vil rutinen dog bestemme funktionen i intervallets yderpunkter.

$\int$'s nøjagtighed

Nøjagtigheden af en funktions integral er afhængig af funktionens nøjagtighed. Derfor er nøjagtigheden af $\int$ afhængig af den nøjagtighed hvorved underprogrammet bestemmer $f(x)$.^{*} De kan selv bestemme, hvor nøjagtigt $f(x)$ skal beregnes, ved at vælge udlæsning med netop så mange cifre, som De mener er nøjagtige.^{**} Hvis De angiver få cifre, vil integralet blive beregnet hurtigt,^{***} men det vil være forudsat, at funktionen faktisk ikke kan angives med større nøjagtighed. Vi vil straks vise, hvorledes De kan bestemme nøjagtigheden af det beregnede integral, men vil først uddybe spørgsmålet om udlæsningsformatets betydning for integrationsrutinen.

De vil erindre, at HP-15C arbejder med tre forskellige udlæsningsformater: $\boxed{\text{FIX}}$, $\boxed{\text{SCI}}$ og $\boxed{\text{ENG}}$. Hvilket af disse formater der skal anvendes, vil i det store og hele afhænge af opgaven og af smag og behag; for de fleste integraler vil De opnå samme resultat, uanset hvilken udlæsningsform der bliver brugt (forudsat der arbejdes med samme antal betydende cifre). Men da det er mest bekvemt at arbejde med $\boxed{\text{SCI}}$'s faste antal cifre, vil vi anvende $\boxed{\text{SCI}}$ i de følgende eksempler.

Bemærk: Husk at så snart De har valgt udlæsningsformat, kan De ændre antallet af udlæste cifre ved at lagre et tal i indeksregistret og derefter bruge $\boxed{f} \boxed{\text{FIX}} \boxed{1}$, $\boxed{f} \boxed{\text{SCI}} \boxed{1}$ eller $\boxed{f} \boxed{\text{ENG}} \boxed{1}$, som omtalt i afsnit 10. Denne mulighed har specielt betydning, når $\int$ er lagret som en programinstruktion.

^{*} Det er muligt, at integraler af funktioner med særlige egenskaber (voldsom oscillation eller diskontinuiteter) bliver beregnet forkert. Men det er ikke særlig sandsynligt. I tillæg E diskuterer vi de generelle karakteristika for funktioner, som under visse omstændigheder kan give problemer.

^{**} En beregnet funktions nøjagtighed afhænger af de enkelte leds kendte nøjagtighed. Hvis f.eks. en fysisk konstant er givet med 3 cifres nøjagtighed, giver det ingen mening at beregne et udtryk, hvor konstanten indgår, med 7 cifres præcision. (Præcision er et udtryk for det antal cifre der regnes med – nøjagtigheden vedrører antallet af korrekte cifre.) Se *HP-15C Advanced Functions Handbook* for en nærmere diskussion af dette emne.

^{***} Baggrunden for dette omtales i tillæg E.

Da nøjagtigheden af ethvert integral er begrænset af nøjagtigheden af funktionen (som angivet af udlæsningsformatet), kan HP-15C ikke beregne et integral eksakt, men kun en *tilnærmelse*. HP-15C lagrer usikkerheden* på integraltilnærmelsen i Y samtidig med at tilnærmelsen lagres i X. De kan således få udlæst usikkerheden med $\boxed{x\geq y}$.

Eksempel: Sæt udlæsningsformatet til $\boxed{\text{SCI}}$ 2 og bestem en tilnærmelse for integralet, der definerer $J_1(1)$ (se eksemplet side 194).

Indtastning	Lyspanel	
$0 \boxed{\text{ENTER}}$	0,0000	Indtast den nedre grænse i Y.
$\boxed{g} \boxed{\pi}$	3,1416	Indtast den øvre grænse i X.
$\boxed{g} \boxed{\text{RAD}}$	3,1416	(Hvis radian ikke allerede er valgt som vinkelmål.)
$\boxed{f} \boxed{\text{SCI}} 2$	3,14 00	Vælger flydende tals repræsentation med 3 betydende cifre.
$\boxed{f} \boxed{\int} 1$	1,38 00	$\boxed{\text{SCI}}$ 2-tilnærmelsen.
$\boxed{x\geq y}$	1,88 - 03	Den tilsvarende usikkerhed.

Integralet er således bestemt til $1,38 \pm 0,00188$. Da usikkerheden først får indflydelse på tilnærmelsens 3. decimal, kan De anse alle udlæste cifre i tilnærmelsen for korrekte. I almindelighed vil det dog være vanskeligt at forudsige, hvor mange af en tilnærmelses cifre der vil være nøjagtige. Det afhænger ganske af den funktion der integreres, af integrationsintervallet og af udlæsningsformatet.

* Man kan naturligvis ikke altid bestemme den eksakte forskel mellem tilnærmelsen og den sande værdi (i så fald var der jo ingen grund til at beregne en tilnærmelse). Men HP-15C beregner en øvre grænse for differensen mellem tilnærmelse og sand værdi. Denne øvre grænse er den udlæste usikkerhed. Når vi f.eks. for $\text{Si}(2)$ får $1,6054 \pm 0,0001$ er tilnærmelsen 1,6054 medens usikkerheden er 0,0001. Det betyder, at om end vi ikke kender den faktiske værdi for $\text{Si}(2)$, ved vi, at det er overordentlig usandsynligt at den ligger uden for intervallet $1,6054 \pm 0,0001$. (Se også første fodnote side 197.)

Hvis usikkerheden er større end De kan acceptere kan De normalt gøre den mindre ved at vælge et større antal betydende cifre under beregningen.*

Når De vil gentage en beregning, er det ikke nødvendigt at genindtaste grænserne. Samtidig med at tilnærmelsen lagres i X og usikkerheden i Y, bliver den øvre grænse lagret i Z og den nedre grænse i T. De skal således blot trykke   for at være klar til en ny beregning af samme bestemte integral.

Eksempel: Bestem nu $J_1(1)$ med fire cifres nøjagtighed i stedet for med kun to.

Indtastning	Lyspanel	
  4	1,8826 - 03	Vælg flydende tal med 5 betydende cifre.
 	3,1416 00	Rul stakken ned indtil den øvre grænse udlæses.
  1	1,3825 00	 4-tilnærmelsen.
	1,7091 - 05	Den tilsvarende usikkerhed.

Usikkerhed angiver at tilnærmelsen er nøjagtig til mindst fire decimaler. Bemærk at usikkerheden på  4-tilnærmelsen er omkring hundrede gange mindre end på  2-tilnærmelsen. I almindelighed kan De regne med at usikkerheden falder med en faktor 10 for hvert ekstra ciffer, der bruges under beregningen.

I det foregående eksempel viste usikkerheden, at tilnærmelsen *måske* kun var korrekt til 4 decimaler. Nu er det en funktion, som faktisk er med et meget stort antal cifre, så vi kan sammenligne tilnærmelsen med den sande værdi. Hvis vi udlæser alle 10 cifre i tilnærmelsen og sammenligner med den sande værdi, vil vi se, at tilnærmelsen er bedre end til 4. decimal. Det betyder således, at usikkerheden er et meget konservativt estimat af uøjagtigheden.

* Dette forudsætter naturligvis, at det giver mening at arbejde med flere cifre. De kan ikke forbedre et resultat ved at indføre tomme cifre.

Indtastning $x \geq y$

f CLEAR PREFIX

Lyspanel**1,3825 00****1382459676**

Tilnærmelsen udlæses igen.

Alle 10 cifre i tilnærmelsen.

Integralet er, med 8 decimaler, lig med 1,38245969. Så tilnærmelsen er nøjagtig til 7. decimal og ikke blot til 4. Men bemærk at der ikke findes nogen metode til at afgøre, om unøjagtigheden er mindre end usikkerheden (hvis der gjorde, ville HP-15C allerede have brugt den metode ...).

I tillæg E giver vi en mere detaljeret gennemgang af nøjagtighed og usikkerhed.

Brug af $\int$ i et program

$\int$ kan bruges som programinstruktion forudsat programmet ikke selv er kaldt af $\int$. $\int$ kan altså ikke bruges rekursivt. Det betyder, at $\int$ ikke kan bruges til at beregne multiple integraler; hvis De alligevel forsøger, vil rutinen stoppe og fejlmeddelelsen **Error 7** bliver udlæst. Men $\int$ kan optræde i et underprogram, som er kaldt af **SOLVE**.

Når De anvender $\int$ i et program, bliver ét af de syv returadresseregistre brugt af rutinen. Samtidig afsluttes det underprogram, som kaldes af $\int$, af en **RTN**-funktion. Det betyder, at der højst kan arbejdes med fem niveauer af underprogramkald, når $\int$ er aktiv under programkørsel. Hvis derimod $\int$ aktiveres fra tastaturet, bliver den første returadresse ikke brugt. Når $\int$ kaldes fra tastaturet, kan der således arbejdes med seks niveauer. Husk at når alle syv registre i returadressestakken er brugt vil forsøg på at kalde et underprogram resultere i **Error 5**. (Se side 104.)

Krav til lagerplads

$\int$ stiller krav om 23 dataregistre. (I tillæg C forklares, hvorledes disse registre automatisk tages fra hukommelsen.) Hvis der ikke er 23 tomme dataregistre til rådighed vil $\int$ ikke køre, og **Error 10** bliver udlæst.

En rutine, som bruger såvel $\int$ som **SOLVE**, stiller krav om 23 ledige dataregistre.

Reference til yderligere information

Dette afsnit har givet Dem al den information, der er nødvendig for at anvende $\int$ i et meget bredt udvalg af anvendelser. I tillæg E, Detaljeret gennemgang af $\int$, gennemgås flere detaljer og anvendelser – blandt andet:

- Hvorledes $\int$ virker.
- Nøjagtighed, usikkerhed og beregningstid.
- Usikkerhed og udlæsningsformat.
- Forhold der kan føre til forkerte resultater.
- Forhold der kan forlænge beregningstiden.
- Udlæsning af foreløbig tilnærmelse.

Tillæg A

Fejlmeddelelser

Hvis De forsøger at udføre en ugyldig operation – f.eks. division med 0 – vil lyspanelet udlæse **Error** og et tal. En sådan fejlmeddelelse og den tilhørende fejltilstand ophæves ved tryk på en vilkårlig tast.

HP-15C har følgende fejlmeddelelser. (Beskrivelsen af **Error 2** omfatter en liste over de formler, der bruges i statistikfunktionerne.)

Error 0: Ugyldige matematiske funktioner

Ugyldigt argument:

$\boxed{\div}$, hvor $x = 0$

$\boxed{y^x}$, hvor:

- $y < 0$ og x ikke heltallig – reelstatus
- $y = 0$ og $x \leq 0$ – reelstatus
- $y = 0$ og $\text{Re}(x) \leq 0$ – kompleksstatus

$\boxed{\sqrt{x}}$, hvor $x < 0$ i reelstatus

$\boxed{1/x}$, hvor $x = 0$

$\boxed{\text{LOG}}$, hvor:

- $x \leq 0$ – reelstatus
- $x = 0$ – kompleksstatus

$\boxed{\text{LN}}$, hvor:

- $x \leq 0$ – reelstatus
- $x = 0$ – kompleksstatus

$\boxed{\text{SIN}^{-1}}$, hvor $|x| > 1$ – reelstatus

$\boxed{\text{COS}^{-1}}$, hvor $|x| > 1$ – reelstatus

$\boxed{\text{STO}} \boxed{\div}$, hvor $x = 0$

$\boxed{\text{RCL}} \boxed{+}$ n , hvor $R_n = 0$

$\boxed{\Delta\%}$, hvor $y = 0$

$\boxed{\text{HYP}^{-1}} \boxed{\text{COS}}$, hvor $x < 1$ – reelstatus

$\boxed{\text{HYP}^{-1}} \boxed{\text{TAN}}$, hvor $|x| > 1$ – reelstatus

$\boxed{\text{P}_{y,x}}$ eller $\boxed{\text{C}_{y,x}}$, hvor:

- x eller y ikke er heltallig
- $x < 0$ eller $y < 0$
- $x > y$
- x eller $y \geq 10^{10}$

Error 1: Ugyldige matrixoperationer

Der er gjort forsøg på anvendelse af andet end en gyldig matrixoperation på en matrix (det adresserede register indeholdt en matrixdeskriptor).

Error 2: Ugyldige statistiske operationer

$$\boxed{\bar{x}} \quad n = 0$$

$$\boxed{s} \quad n \leq 1$$

$$\boxed{\hat{y}, r} \quad n \leq 1$$

$$\boxed{\text{L.R.}} \quad n \leq 1$$

Error 2 udlæses desuden, hvis der divideres med nul eller uddrages kvadratrods af et negativt tal under beregning af en af følgende formler:

$$\bar{x} = \frac{\sum x}{n}$$

$$\bar{y} = \frac{\sum y}{n}$$

$$s_x = \sqrt{\frac{M}{n(n-1)}}$$

$$s_y = \sqrt{\frac{N}{n(n-1)}}$$

$$r = \frac{P}{\sqrt{M \cdot N}}$$

$$A = \frac{P}{M}$$

$$B = \frac{M \sum y - P \sum x}{n \cdot M}$$

$$\hat{y} = \frac{M \sum y + P(n \cdot \bar{x} - \sum x)}{n \cdot M}$$

hvor:

$$M = n \sum x^2 - (\sum x)^2$$

$$N = n \sum y^2 - (\sum y)^2$$

$$P = n \sum xy - \sum x \sum y$$

(A og B er resultaterne af

$\boxed{\text{L.R.}}$, hvor $y = Ax + B$)

Error 3: Ugyldig registeradresse eller matrixelement

Der er gjort forsøg på at adressere et ikke-eksisterende register eller matrixelement.

Error 4: Ugyldigt linienummer eller ugyldig etiket

Det kaldte linienummer er enten for stort (> 448) eller ikke i brug, eller der er gjort forsøg på at lagre et for stort program, eller den kaldte etiket eksisterer ikke.

Error 5: For mange niveauer af underprogrammer

Der er kaldt underprogrammer til mere end 7. niveau (5. med **SOLVE** og/eller $\int$).

Error 6: Ugyldigt flagnummer

Der er gjort forsøg på at arbejde med et flag, hvis nummer er større end 9.

Error 7: Rekursiv brug af **SOLVE eller $\int$**

Et underprogram, som er kaldt af **SOLVE**, indeholder selv **SOLVE**, eller et underprogram, som er kaldt af $\int$, indeholder selv $\int$.

Error 8: Ingen rod

SOLVE fandt ingen rod med de givne startværdier.

Error 9: Service

Selvtesten opdagede en kredsløbsfejl eller der blev trykket på en forkert tast under selvtesten. Se tillæg F.

Error 10: Utilstrækkelig hukommelse

Der er ikke tilstrækkelig plads i hukommelsen til at udføre den ønskede operation.

Error 11: Ugyldigt matrixargument

Et eller begge matrixargumenter er logisk forkerte eller på anden måde ugyldige:

$\boxed{+}$ eller $\boxed{-}$, hvor dimensionerne ikke passer sammen

$\boxed{\times}$, hvor:

- dimensionerne ikke passer sammen
- resultatmatricen er den samme som en af argumentmatricerne

$\boxed{1/x}$, hvor matricen ikke er kvadratisk.

Skalar/matrix $\boxed{+}$, hvor matricen ikke er kvadratisk.

$\boxed{\div}$, hvor:

- matricen i X ikke er kvadratisk
- dimensionerne ikke passer sammen
- resultatmatricen er samme matrix som lagret i X

MATRIX 2, hvor inddata er en skalar eller hvor antallet af rækker er ulige.

MATRIX 3, hvor inddata er en skalar eller hvor antallet af søjler er ulige.

MATRIX 4, hvor inddata er en skalar.

MATRIX 5, hvor:

- inddata er en skalar
- dimensionerne ikke passer sammen
- resultatmatricen er den samme som en af argumentmatricerne

MATRIX 6, hvor:

- inddata er en skalar
- dimensionerne (inkl. resultatmatricens) ikke passer sammen
- resultatmatricen er den samme som en af argumentmatricerne

MATRIX 9, hvor matricen ikke er kvadratisk.

RCL **DIM** $\boxed{1}$, hvor indholdet af R_1 er en skalar.

DIM $\boxed{1}$, hvor indholdet af R_1 er en skalar.

STO **RESULT**, hvor inddata er en skalar.

$\boxed{P_{y,x}}$, hvor antallet af søjler er ulige.

$\boxed{C_{y,x}}$, hvor antallet af rækker er ulige.

Pr Error (Power Error)

Den kontinuerlige hukommelse har været afbrudt og alle registre er blevet nulstillet.

Tillæg B

Stakløft og SIDSTE X

HP-15C er konstrueret, så den fungerer så naturligt som muligt. Som De allerede har set under gennemgangen af denne håndbog, behøver De sjældent tænke på, hvordan den automatiske rullestak virker.

Der er dog nogle tilfælde, særlig under programmering af lommeregneren, hvor det er nødvendigt at have fuldt overblik over rullestakkens indhold og adfærd. Til det formål skulle følgende kunne være Dem til hjælp.

Afslutning af talindtastning

De fleste operationer vil, uanset om de udføres fra tastaturet eller under programkontrol, afslutte en talindtastning. Det betyder i praksis, at maskinen vil opfatte det første ciffer efter alt andet end en talindtastning som første ciffer i et nyt tal.

De eneste operationer der *ikke* afslutter talindtastning er taltasterne:



Stakløft

HP-15C's operationer kan opdeles i tre grupper alt efter deres indvirken på stakløften: de kan *aktivere* stakløften, *blokere* den eller være *neutrale*.

Når HP-15C er i kompleksstatus vil alle operationer virke på såvel reel- som imaginærstakken. Stakløften virker på samme måde på begge stakke. Dertil kommer, at tallet som indtastes i det reelle X-register (lyspanelregistret) *efter enhver anden operation end*  eller  vil lagre 0 i imaginærstakkens X-register.

Operationer, der blokerer stakløften

Stakløft. Når stakløften er blokeret, vil det næste tal *overskrive* indholdet

af X-registret uden at indholdet af de øvrige stakregistre påvirkes. Kun fire operationer blokerer stakløften*:

[ENTER] [CLx] [$\Sigma+$] [$\Sigma-$]

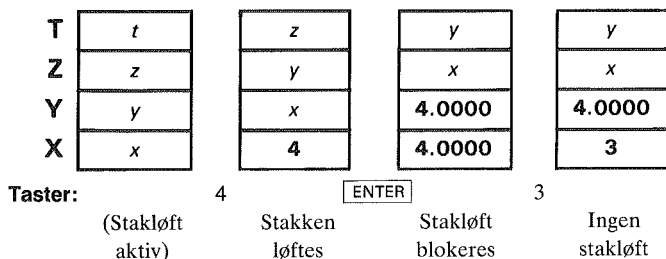
Imaginære X-register. Det imaginære X-register nulstilles samtidigt med lagring i det reelle X-register efter brug af [ENTER], [$\Sigma+$] eller [$\Sigma-$]. Derimod nulstilles det imaginære X-register *ikke* efter $\leftarrow$ eller [CLx].

Operationer, der aktiverer stakløften

Stakløft. Når stakløften er aktiveret, vil indtastning af et tal bevirke, at det gamle indhold af X-registret skubbes op i Y-registret (hvis indhold skubbes op i Z, som skubbes op i T – hvis gamle indhold går tabt).

Langt de fleste operationer aktiverer stakløften – det gælder f.eks. alle de matematiske standardfunktioner. Bemærk, at skift fra programmeringsstatus til normalstatus aktiverer stakløften. Når der tændes for maskinen, aktiveres stakløften.

(Husk, at når X-registret er skraveret i et af de følgende diagrammer, betyder det, at dets indhold vil blive overskrevet af det næste indtastede eller fremkaldte tal.)



* Se fodnote side 36.

T	y	y	y	y
Z	x	x	x	x
Y	4.0000	53.1301	53.1301	53.1301
X	3	5.0000	0.0000	7
Taster:	g	g P	g CLx	7
		Stakken aktiveres	Stakken blokeres	Ingen stakløft

Imaginære X-register. Alle aktiverende funktioner nulstiller det imaginære X-register, når det næste tal indtastes eller fremkaldes.

Neutrale operationer

De neutrale operationer påvirker ikke stakløften; den er således i samme stand før og efter udførelsen af sådanne funktioner. De neutrale operationer er typisk funktioner, der ændrer maskinens status men i øvrigt ikke udfører nogen funktion på tal i rullestakken.

Hvis man f.eks. har trykket **ENTER** og dernæst ændrer fastkomma med **FIX**, vil stakløften stadig være blokeret. Også de taster, der bruges til talindtastning, er i sagens natur neutrale – dog vil **CHS** og **EEX** aktivere stakløften, hvis de ikke bruges i umiddelbar forlængelse af en talindtastning. **◀** blokerer stakløften, når den bruges efter afslutning af talindtastning.

Følgende operationer er neutrale:

FIX	GRD	USER	R/S
SCI	GTO CHS <i>nnn</i>	CLEAR PREFIX	P/R
ENG	BST	CLEAR REG	(i)*
DEG	SST	CLEAR Σ	
RAD	MEM	PSE	

Imaginære X-register. De ovennævnte funktioner er også neutrale med hensyn til nulstilling af det imaginære X-register.

* Når **f** **(i)** bruges til udlæsning af det imaginære X-register.

SIDSTE X

Følgende operationer lagrer sidste x-værdi i SIDSTE X:

-	x^2	HYP ⁻¹	COS	%
+	SIN	HYP ⁻¹	TAN	$\Delta\%$
$\times$	COS	$\Rightarrow$ H.MS	$\Rightarrow$ P	
$\div$	TAN	$\Rightarrow$ H	$\Rightarrow$ R	
ABS	SIN ⁻¹	$\Rightarrow$ DEG	$P_{y,x}$ *	
FRAC	COS ⁻¹	$\Rightarrow$ RAD	$C_{y,x}$ *	
INT	TAN ⁻¹	LN	$\Sigma+$	
RND	HYP SIN	e^x	$\Sigma-$	
$1/x$	HYP COS	LOG	$\oint, r$	
$x!$	HYP TAN	10^x	MATRIX 5 til 9	
$\sqrt{x}$	HYP ⁻¹ SIN	y^x	$\oint$ **	

* Undtagen i forbindelse med matricer.

** $\oint$ bruger SIDSTE X på en speciel måde – se tillæg E.

Tillæg C

Splitning

Hukommelsen

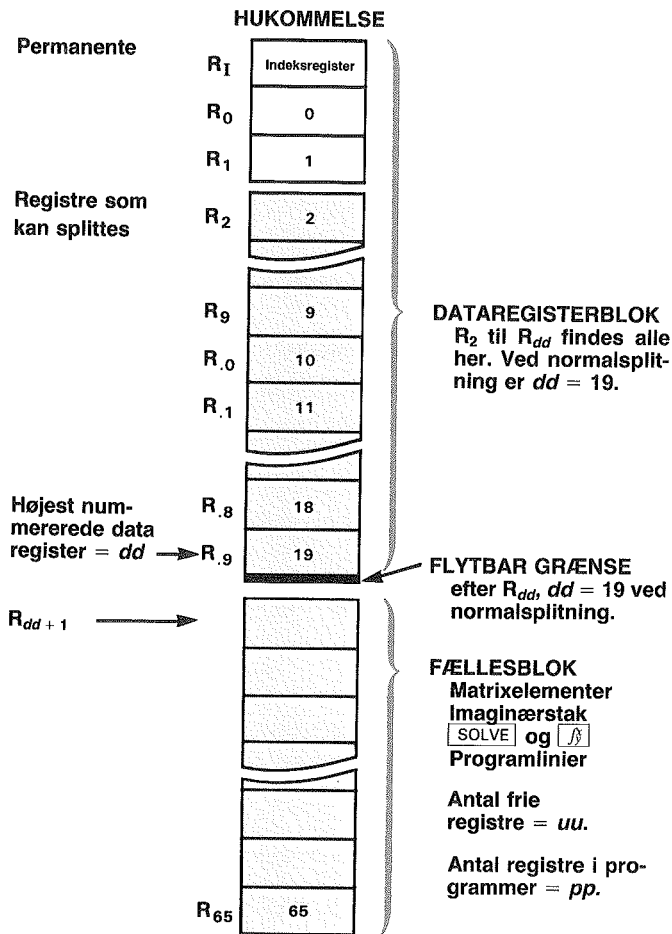
Lagring i dataregistre, programlinier og udførelse af avancerede funktioner* trækker alle på samme basishukommelse i HP-15C. Det er *splitningen* (og den samlede kapacitet), der afgør om der er tilstrækkelig lagerplads til *rådighed* for en bestemt funktion.

Registre

Hukommelsen i HP-15C er opdelt i registre, som er samlede i *registerblokke*. Disse blokke har helt veldefinerede anvendelser. I blokkene vil der til enhver tid være i alt 67 registre.

- *Dataregisterblokken* indeholder registre, som udelukkende kan bruges til lagring af data. Når den kontinuerlige hukommelse har været nulstillet, er der 21 registre i denne blok, som altid indeholder mindst 3 registre: R_1 , R_0 og R_{-1} .
- *Fællesblokken* indeholder registre, som frit kan benyttes til programmer, matricer, imaginærstak samt $\boxed{\text{SOLVE}}$ og $\boxed{\frac{\partial}{\partial x}}$. Efter nulstilling af den kontinuerlige hukommelse er der således 46 frie registre i fællesblokken.

* $\boxed{\text{SOLVE}}$, $\boxed{\frac{\partial}{\partial x}}$, kompleksstatus og matricer kræver alle, at der afsættes ekstra lagerplads – se mere herom senere i dette tillæg.



Splitbar hukommelse: 64 registre som er nummereret fra R_2 til R_{65} . $[(dd - 1) + uu + pp + (\text{matricelementer}) + (\text{imaginærstak}) + (\text{SOLVE } \beta)] = 64$. Ved splitting og indirekte adressering bliver registre R_0 til R_9 refereret til som R_{10} til R_{19} .

Udlæsning af aktuel splitning med MEM

De kan få udlæst den aktuelle splitning ved at trykke på 9 MEM og holde MEM nedtrykket.* Der bliver udlæst i alt fire tal:

dd uu pp-b

hvor:

dd = nummeret på det *højstnummererede* register i dataregisterblokken.
(Antallet af dataregistre bliver således $dd + 2$ – idet R_0 og R_1 ellers ikke medregnes.)

uu = antallet af frie registre i fællesblokken.

pp = antallet af registre som indeholder programinstruktioner.

b = antallet af ubrugte bytes i det højstnummererede register som indeholder programinstruktioner.

Efter nulstilling af den kontinuerlige hukommelse vil 9 MEM udlæse:

19 46 0-0

Den flytbare grænse mellem de to blokke befinder sig altid mellem R_{dd} og R_{dd+1} .

Splitning

Hukommelsen består af i alt 67 registre på hver 7 bytes. 64 af disse registre (R_2 til R_{65}) kan indgå i såvel dataregisterblokken som fællesblokken.

Brugen af DIM (i)

Hvis De får brug for mere fælleslager (til f.eks. programmering) eller for mere plads til data (men ikke begge dele samtidig!) kan De udføre den nødvendige omfordeling af registre med DIM (i).** Proceduren er således:

* 9 MEM er ikke programmerbar.

** DIM (*dimension*) har sit navn fordi den normalt bruges med A – E eller I til at dimensionere matricer. Men den kan desuden bruges med (i) til at dimensionere de to registerblokke.

1. Indtast *dd*, nummeret på det højstnummerede dataregister De ønsker. $1 \leq dd \leq 65$. Antallet af registre i fællesblokken (som er potentielt ledige til brug for programinstruktioner) bliver derved $(65 - dd)$.
2. Tryk **f** **DIM** **(i)**.

Splitningen kan inspiceres på to måder:

- Tryk **RCL** **DIM** **(i)** så registernummeret for det højstnummerede register (*dd*) bliver udlæst.
- Tryk **g** **MEM** (som omtalt ovenfor) der udlæser en mere fyldig splitningsbeskrivelse (*dd uu pp-b*).

Indtastning

Lyspanel

(programhukommelsen forudsættes at være slettet)*

1 f DIM (i)	1,0000	R_1, R_0 og R_1
g MEM (hold)	1 64 0-0	udlagt som dataregistre. 64 registre er frie; ingen indeholder programinstruktioner.
19 f DIM (i)	19,0000	R_{19} (R_9) er det højstnummerede
RCL DIM (i)	19,0000	dataregister. Der er stadig 46 registre i fællesblokken.

Grænser for splitningen

Den kontinuerlige hukommelse vil fastholde splitningen indtil denne ændres med **DIM** **(i)** eller indtil den kontinuerlige hukommelse nulstilles. Hvis De prøver at splitte med en registerstørrelse som er mindre end 1, bliver *dd* sat til 1. Hvis De prøver at splitte med en registerstørrelse som er større end 65, udløses en fejltilstand med fejlmeddelelsen **Error 10**.

* Hvis programhukommelsen ikke slettes, vil antallet af frie registre (*uu*) være mindre. *pp* ville således være større end 0 og *b* variere.

Bemærk følgende når De konverterer registre:

- Fra fællesblokken kan De kun konvertere *frie* registre. Hvis De f.eks. forsøger at konvertere programlinier til dataregistre, udløses fejltilstanden med fejlmeddelelsen **Error 10**.
- Fra dataregisterblokken kan De konvertere ethvert dataregister; men det nulstilles ved konverteringen. **Error 3** fremkommer, hvis De forsøger at adressere et konverteret register (som jo ikke længere eksisterer). De bør derfor altid starte lagring fra de første registre, således at dataregistre med de højeste numre uden videre kan konverteres, hvis der bliver brug for det.

Programhukommelse

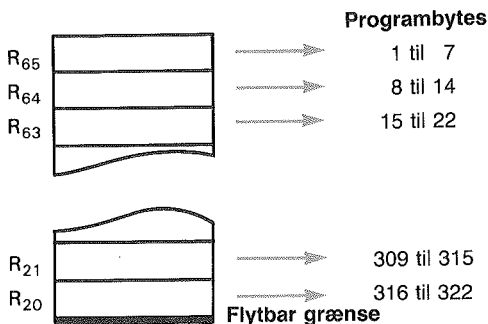
Som tidligere omtalt består hvert register af 7 bytes. Hver instruktion optager 1 eller 2 bytes. De fleste programlinier vil bestå af en enkelt byte; de der består af 2 bytes er opført på side 215.

HP-15C's *maksimale* programmeringskapacitet er på 448 bytes (64 konvertible registre à 7 bytes). Efter nulstilling af den kontinuerlige hukommelse vil der være 322 programbytes til rådighed (46 registre à 7 bytes).

Automatisk splitning

Inden for fællesblokken udføres en automatisk splitning således, at programhukommelsen udvides i takt med behovet. Det gøres ved at der konverteres et enkelt frit fællesregister ad gangen. Programmer lagres fra det højstnummererede fri register.

Konvertering af fri fællesregistre til programregistre



Det første program vil blokere alle 7 bytes i R₆₅. Den 8. byte vil blive taget fra R₆₄ osv. indtil den flytbare grænse nås. Registerne fra dataregisterblokken kan ikke bruges uden ny splitning med **DIM** **(i)**.

Programinstruktioner på 2 bytes

De følgende instruktioner er de eneste, som optager to bytes i programhukommelsen. (Alle andre optager kun én.)

f **LBL** **.** etiket
f **GTO** **.** etiket
g **CF** (n eller **I**)
g **SF** (n eller **I**)
g **F?** (n eller **I**)
f **FIX** (n eller **I**)
f **SCI** (n eller **I**)
f **ENG** (n eller **I**)
f **SOLVE**
f **$\frac{\square}{\square}$**

f **MATRIX** {0 til 9}
f **$x \geq$** {2 til 9, .0 til .9}
f **DSE** {2 til 9, .0 til .9}
f **ISG** {2 til 9, .0 til .9}
STO {+, -, ×, ÷}
RCL {+, -, ×, ÷}
STO **MATRIX** {**A** til **E**}
STO {**A** til **E**, **(i)**} i USER-status
RCL {**A** til **E**, **(i)**} i USER-status
STO **g** **(i)**
RCL **g** **(i)**

De avancerede rutiners pladskrav

De fire avancerede rutiner stiller krav om registre, som tages fra fælles-registrene.

Funktion	Antal nødvendige registre
SOLVE $\frac{\square}{\square}$	5 23 } 23 hvis brugt sammen
Kompleks stak	5
Matricer	1 pr. matricelement

SOLVE og **$\frac{\square}{\square}$** styrer selv den nødvendige udlægning af registre.* Det betyder, at funktionerne kun optager plads, når de bruges.

* Hvis De afbryder **SOLVE** eller **$\frac{\square}{\square}$** medens rutinen arbejder, kan De frigøre de udlagte registre ved at trykke **g** **RTN** eller **f** **CLEAR** **PRGM** i normalstatus.

Der afsættes plads til den imaginære stak, når `f` `I`, `f` `Re≥Im` eller `g` `SF` 8 bruges. Den imaginære stak opløses af `CF` 8.

Der afsættes ikke plads til matricer før de dimensioneres (med `DIM`). Når en matrix redimensioneres afsættes det antal registre, der svarer til den nye dimension. Med `MATRIX` 0 får alle matricer dimensionen 0×0 .

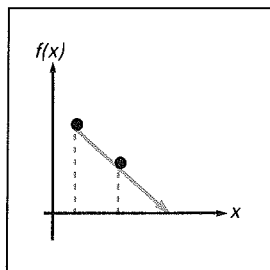
Detaljeret gennemgang af **SOLVE**

I afsnit 13, Løsning af ligninger, kan De finde alle oplysninger, som er nødvendige for at bruge **SOLVE** effektivt. I dette tillæg kan De desuden finde mere avanceret information og yderligere diskussion af **SOLVE**.

Mekanikken bag **SOLVE**

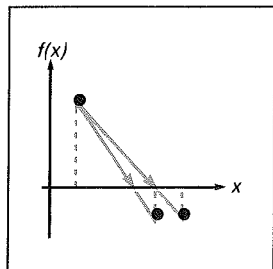
De vil få mest ud af **SOLVE**, hvis De ved hvorledes algoritmen virker.

Under søgningen efter et nulpunkt for den givne funktion vil algoritmen bruge funktionsværdien for de to eller tre seneste estimater til at tilnærme funktionens graf. Algoritmen anvender denne form til et intelligent "gæt" på et nyt estimat for skæringspunktet på x -aksen. Herefter udføres funktionsunderprogrammet, så funktionsværdien for det nye estimat beregnes. Denne proces udføres gentagne gange af **SOLVE**-algoritmen.



Hvis to estimater giver funktionsværdier med modsat fortegn antager algoritmen, at der findes mindst et nulpunkt i intervallet mellem disse estimater. Intervallet indsnævres herefter systematisk indtil der findes en rod.

Algoritmen betragter søgningen som fuldført, hvis den beregnede funktionsværdi er nul eller hvis funktionsværdierne for to estimater, som kun afviger fra hinanden med en enhed på sidste betydende ciffer, har modsat fortegn. I disse tilfælde indstilles beregningen, og det aktuelle estimat bliver udlæst.

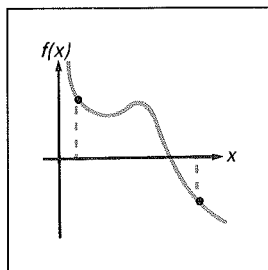


Som omtalt i afsnit 13, side 184, vil der sandsynligvis ikke være noget nulpunkt, hvis ingen af disse situationer optræder. Det skyldes normalt, at rutinen ikke har mulighed for en logisk bestemmelse af et nyt estimat, som ligger nærmere nul. Denne situation markeres med **Error 8**.

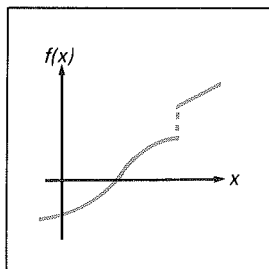
De bør bemærke, at startværdierne (som De indtaster) bliver brugt under den iterative "gætte"proces. Da gode startværdier giver mulighed for nøjagtigere bestemmelser af nye estimater kan velvalgte startværdier lette nulpunktssøgningen ganske betydeligt.

SOLVE vil *altid* finde en rod, hvis der eksisterer en inden for maskinens talområde og mindst én af følgende fire betingelser er opfyldt:

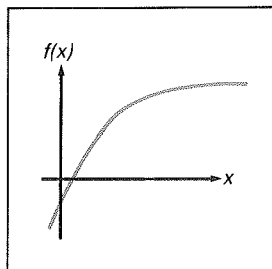
- To estimater har funktionsværdier med modsat fortegn.



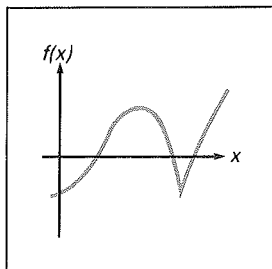
- Funktionen er monoton, dvs. at $f(x)$ altid enten vokser eller falder med voksende x .



- Funktionen graf er overalt konveks eller konkav.



- Mellem hvert rodpar findes højst ét lokalt minimum eller maksimum.



Det forudsættes i alle tilfælde, at SOLVE ikke afbrydes på grund af ugylde operationer eller eksterne forhold.

Nøjagtighed

Når De bruger SOLVE til at finde en lignings rødder, findes disse med en nøjagtighed, der svarer til maskinens nøjagtighed. En udlæst rod vil enten have en funktionsværdi, som er eksakt nul eller være det 10-cifrede tal, som ligger roden nærmest. Enhver sådan rod vil have en fejl som højst er 2-3 enheder på 10. betydende cifre.

I de fleste tilfælde vil den fundne rod være et nøjagtigt estimat af den teoretiske (uendeligt præcise) rod. Men under visse betingelser vil maskinens endelige antal cifre føre til resultater, som ikke passer med Deres teoretiske forventninger.

Hvis en beregning resulterer i et tal, hvis numeriske værdi er mindre end $1,000000000 \times 10^{-99}$ vil resultatet blive sat til nul. Denne effekt omtales som "underløb". Hvis underprogrammet, som beregner funktionen, giver underløb for et x -interval og hvis dette får indflydelse på funktionens værdi, så er der grund til at forvente en vis unøjagtighed for enhver rod i dette interval. F.eks. har funktionen

$$x^4 = 0$$

roden $x = 0$. Men på grund af underløb vil SOLVE stoppe ved **1,5060 - 25** (når 1 og 2 bruges som startværdier). Et andet eksempel er

$$1/x^2 = 0$$

som ikke har nogen rod. På grund af underløb under beregninger vil SOLVE alligevel give **3,1707 49** som rod (med 10 og 20 som startværdier). I begge disse eksempler har rutinen fundet en værdi af x , som resulterer i en funktionsværdi, som sættes til 0 af maskinen. Når blot De er opmærksom på disse forhold, skulle De uden besvær kunne fortolke sådanne resultater korrekt.

Nøjagtigheden af beregnede rødder kan i nogle tilfælde blive påvirket af de afrundingsfejl, som uundgåeligt indtræder når der arbejdes med et endeligt antal cifre. Hvis underprogrammet forudsætter præcision ud over 10 betydende cifre for at kunne beregne funktionsværdier for argumenter i et givet interval, bliver SOLVE-resultaterne måske unøjagtige. F.eks. har ligningen

$$|x^2 - 5| = 0$$

$\sqrt{5}$ som rod. Da der ikke findes noget 10-cifret tal, som er eksakt lig med $\sqrt{5}$ og da funktionen aldrig bliver nul eller skifter fortegn, vil SOLVE give fejlmeddelelsen **Error 8** for ethvert valg af startværdier. På den anden side vil f.eks.

$$[(|x| + 1) + 10^{15}]^2 = 10^{30}$$

ikke have nogen rødder – fordi ligningens venstreside altid er større end højresiden. Men på grund af afrundingsfejl ved beregningen af

$$f(x) = [(|x| + 1) + 10^{15}]^2 - 10^{30}$$

vil [SOLVE] give løsningen **1,0000** med 1 og 2 som startværdier. Hvis De på forhånd gør Dem klart på hvilken måde [SOLVE] påvirkes af afrundingsfejl, vil De ofte kunne omskrive funktionen således, at afrundingsfejlenes effekt reduceres eller elimineres.

I en lang række funktioner, der refererer til virkeligheden, er nogle af parametrene – og måske hele funktionen – tilnærmelser. Fysiske parametre er behæftet med en uundgåelig måleusikkerhed. Matematiske repræsentationer af fysiske processer er alene modeller og er som sådanne højst så nøjagtige som de forestillinger og tilnærmelser de er baserede på. Også her gælder det, at det kan svare sig at overveje effekten af sådanne unøjagtigheder. De vil f.eks. kunne spare tid, hvis De indlægger en korrektionsmekanisme, som nulsætter beregnede funktionsværdier der er nær nul. I de eksempler vi gennemgik i forbindelse med underløb var der netop tale om, at funktionsværdier i lang tid og under mange gennemløb blot nærmede sig nul ganske langsomt.

Eksempel: Boldmester Castholm og hans lige kan kaste en bold til 105 meters højde eller mere. Det er faktisk sådan, at Castholms bolde normalt når op til 107 meter. Hvor lang tid tager det for en af hans bolde at nå 107 meters højde, når han kaster som beskrevet side 182 i afsnit 13?

Løsning: Den ønskede løsning er værdien af t for $h = 107$. Indtast underprogrammet fra side 182, som beregner højden af kastet. Dette underprogram kan bruges i en ny funktion til at beregne

$$f(t) = h(t) - 107.$$

Følgende underprogram beregner $f(t)$:

Indtastning

g	P/R
f	LBL B
GSB	A
1	
0	
7	
-	
g	RTN

Lyspanel

000 –	Programmeringsstatus.
001 – 42.21.12	Start med ny etiket.
002 – 32 11	Beregn $h(t)$.
003 – 1	
004 – 0	
005 – 7	Beregn $h(t) - 107$.
006 – 30	
007 – 43 32	

For at sikre, at vi finder det første tidspunkt, hvor højden er 107 meter, bruger vi 0 og 1 som startværdier og udfører SOLVE med B.

Indtastning	Lyspanel	
g P/R		Normalstatus.
0 ENTER	0,0000	} Startværdier.
1	1	
f SOLVE B	4,1718	Den ønskede rod.
R ↓	4,1718	Foregående estimat.
R ↓	0,0000	Værdi af $f(t)$ for roden.

Det tager bolden 4,1718 sekunder at nå en højde på præcis 107 meter. (Det tager HP-15C ca. 1 minut at nå dette resultat.)

Lad os nu antage, at funktionen $h(t)$ kun er givet med 1 meters nøjagtighed. Det betyder, at De kan ændre underprogrammet således, at $f(t)$ sættes til nul, når blot den numeriske værdi af $f(t)$ beregnes til mindre end 0,5 meter. Underprogrammet skal ændres således:

Indtastning	Lyspanel	
g P/R	000 -	Programmeringsstatus.
GTO CHS 006	006 - 30	Linien før RTN .
g ABS	007 - 43 16	Numerisk værdi af $f(t)$.
.	008 - 48	} Nøjagtighed.
5	009 - 5	
g TEST 7	010 - 43.30. 7	} Test for $x > y$. Erstat med 0 hvis $0,5 > f(t) $.
g CLx	011 - 43 35	
g TEST 0	012 - 43.30. 0	} Test for $x \neq 0$. $f(t)$ retableres, hvis $x \neq 0$.
g LSTx	013 - 43 36	

Udfør SOLVE igen:

Indtastning

g P/R
 0 ENTER
 1
f SOLVE B
R↓
R↓

Lyspanel

0,0000
1
4,0681
4,0681
0,0000

Normalstatus.

Startværdier

Den ønskede rod.

Foregående estimat.

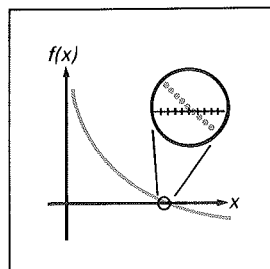
Værdien af $f(t)$ for den modificerede funktion.

Efter 4,0681 sekunder når bolden en højde på $107 \pm 0,5$ meter. Selv om løsningen er forskellig fra den foregående er den korrekt under hensyn til usikkerheden på højden. (Denne løsning findes på mindre end et halvt minut til trods for at der nu er indlagt testoperationer.)

Fortolkning af resultater

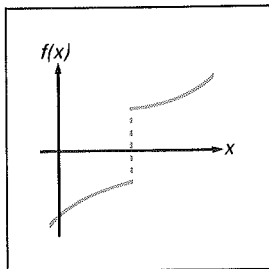
De tal, som SOLVE lagrer i X, Y og Z, kan bruges til at vurdere resultatet af nulpunktssøgningen.* Selv når der ikke findes en rod, kan resultater have væsentlig betydning.

Når SOLVE finder en rod for en specificeret ligning, bliver roden og den tilhørende funktionsværdi lagret i henholdsvis X og Z. En funktionsværdi på nul er det forventede resultat. Men også en funktionsværdi, som er forskellig fra nul, vil kunne accepteres, da den antyder, at funktionens graf skærer x-aksen ganske nær den beregnede rod. I de fleste sådanne tilfælde vil funktionsværdien være relativt nær nul.

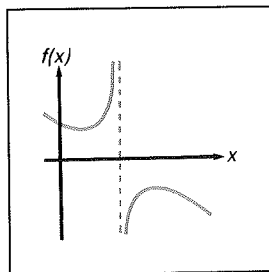


* Tallet i T er det samme, som blev lagret i Y efter sidste gennemkørsel af funktionsunderprogrammet. I almindelighed er dette tal ikke af interesse.

Specielle overvejelser og hensyn er nødvendige i de tilfælde, hvor SOLVE finder en rod, som har en funktionsværdi, der er forskellig fra nul. Hvis funktionen har en diskontinuitet, som skærer x -aksen (således at x er et *springpunkt*), vil HP-15C opgive en rod, som ligger nær diskontinuiteten. Det er en rimelig løsning, da en meget stor ændring i funktionsværdien for to meget nærtliggende x -værdier kunne være resultatet af en hurtig, men kontinuert ændring. Da algoritmen ikke har mulighed for at skelne mellem disse tilfælde, bliver roden udlæst, så De selv kan fortolke resultatet.



Et andet tilfælde, der kan give problemer, gælder funktioner som er stykkevis kontinuerte, med et enkelt punkt, hvor funktionsværdien går mod uendeligt. Hvis funktionsværdien skifter fortegn i diskontinuitetspunktet, bliver dette opfattet som et springpunkt og vælges derfor som rod. Men funktionsværdien, som lagres i Z , vil i disse tilfælde være numerisk relativt meget større end nul.



Hvis diskontinuiteten indtræffer i et punkt, som er givet eksakt med 10 betydende cifre, vil SOLVE formodentlig bruge diskontinuitetspunktet og beregner derfor funktionsværdien – en beregning, som naturligvis fører til overløb og stop af programudførelsen. I sådanne tilfælde kan De alligevel nå vidt i Deres analyse af funktionen ved at indsætte skønsomt valgte testoperationer.

Eksempel: Ingeniør Elisa Trosse arbejder med jernbeton og har fundet, at forskydningsspændingen kan udtrykkes som

$$Q = \begin{cases} 3x^3 - 45x^2 + 350 & \text{for } 0 < x < 10 \\ 1000 & \text{for } 10 \leq x < 14 \end{cases}$$

hvor Q er forskydningsspændingen i N/m^2 og x er afstanden i meter fra den ene ende.



Skriv et underprogram som beregner forskydningsspændingen for en vilkårlig værdi af x . Brug dernæst SOLVE til at finde det punkt, hvor spændingen er nul.

Løsning: Ligningen for forskydningsspændingen for x mellem 0 og 10 kan med fordel omskrives efter Horner's metode:

$$Q = (3x - 45)x^2 + 350 \quad \text{for } 0 < x < 10.$$

Indtastning

g P/R
f LBL 2
 1
 0
g $x \leq y$
GTO 9
g CLx
 3
x
 4
 5
-
x
x
 3
 5
 0
+
g RTN
f LBL 9

Lyspanel

000 -		Programmeringsstatus.
001 - 42.21. 2		
002 - 1	}	Test x -interval.
003 - 0		
004 - 43 10		
005 - 22 9		
006 - 43 35		Hop ved $x \geq 10$.
007 - 3		
008 - 20		$3x$.
009 - 4		
010 - 5		
011 - 30		$(3x - 45)$.
012 - 20		
013 - 20		$(3x - 45)x^2$.
014 - 3		
015 - 5		
016 - 0		
017 - 40		$(3x - 45)x^2 + 350$.
018 - 43 32		Slut på underprogram.
019 - 42.21. 9		Underprogram for $x \geq 10$.

IndtastningEEX

3

g RTN**Lyspanel****020 – 26****021 – 3****022 – 43 32** $10^3 = 1000$.

Slut på underprogram.

Udfør nu SOLVE med 7 og 14 som startværdier, så der søges fra søjlens yderkant.

Indtastningg P/R7 ENTER

14

f SOLVE 2R↕ R↕**Lyspanel****7,0000****14****10,0000****1.000,0000**

Normalstatus.

} Startværdier.

Mulig løsning.

Spændingen er ikke nul.

Den høje spændingsværdi ved den fundne rod viser, at SOLVE har fundet en diskontinuitet; det sted på søjlen, hvor spændingen skifter meget brat fra negativ til positiv. De bør derfor starte fra den anden ende og f.eks. bruge startværdierne 0 og 7.

Indtastning0 ENTER

7

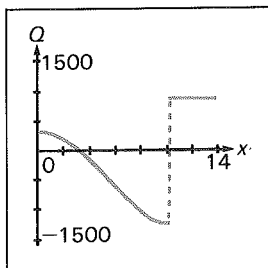
f SOLVE 2R↕ R↕**Lyspanel****0,0000****7****3,1358****2,0000 – 07**

} Startværdier.

Mulig løsning.

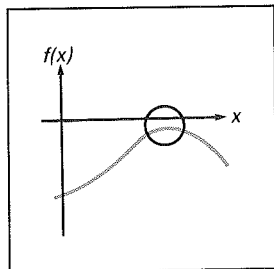
Negligabel spænding.

Trosse's søjle har en forskydningsspænding på nul ca. 3,1358 meter fra den ene ende og et kraftigt skift i spænding 10,0000 meter fra enden.

Grafen for Q mod x

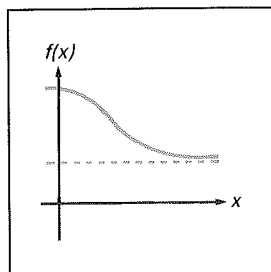
Når der ikke findes nogen rod og **Error 8** bliver udlæst, kan De bruge ◀ eller en anden tast til at slette fejlmeddelelsen og se, hvilket estimat der gav så lille en funktionsværdi som muligt. Ved også at betragte tallene i Y og Z kan De i mange tilfælde bestemme funktionens adfærd nær rod-estimatet og anvende denne information på en konstruktiv måde.

Hvis algoritmen indstiller sin søgen nær et punkt, hvor funktionen numerisk har et lokalt minimum, skal De ophæve **Error 8**-udlæsningen og gennemgå stakregistre. Hvis indholdet af Z er relativt nær nul, er det muligt at der faktisk er fundet en rod – således at tallet i X er det 10-cifrede tal, der kommer nærmest den faktiske rod. De kan undersøge dette potentielle minimum nærmere ved at rulle stakken indtil den er retableret og derefter starte SOLVE med de sidst beregnede estimater som startværdier. Hvis der er fundet et ægte lokalt minimum vil **Error 8** igen blive udlæst og tallet i X vil være meget nær det foregående, men måske nok nærmere det lokale minimum.

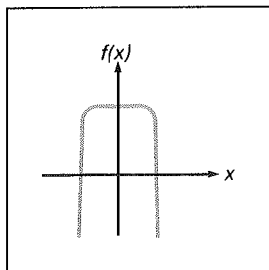


De kan naturligvis bruge denne effekt til at søge efter lokale, numeriske minima. Men De skal i så fald sørge for at begrænse søgningen til et interval nær om det ønskede lokale, numeriske minimum. Husk, at SOLVE virkelig gør meget for at finde et nulpunkt.

Hvis algoritmen stopper med **Error 8** fordi den arbejder sig ud ad en vandret asymptote (hvor funktionsværdierne er stort set uændrede i et stort x -interval), vil estimaterne i X og Y sædvanligvis være meget forskellige. Tallet i Z er værdien for den potentielle asymptote. Hvis De udfører SOLVE igen med X- og Y-registrenes indhold som startværdier, vil De formodentlig igen få **Error 8**, hvis der er tale om en vandret asymptote, men i så fald vil tallene i X og Y være forskellige fra de foregående. Indholdet af Z vil ikke have ændret sig.



Hvis **Error 8** udlæses som resultat af, at en søgning koncentrerer sig om et lokalt "fladt" interval, vil estimerterne i X og Y være relativt nær hinanden eller ekstremt små. Udfør SOLVE igen med tallene i X og Y som startværdier (eller eventuelt nogle, der ligger lidt længere fra hinanden). Hvis funktionen numerisk hverken er meget nær et minimum eller konstant, vil SOLVE i sidste ende udvide sit søgeinterval og finde et mere betydeligt resultat.



Eksempel: Undersøg adfærden af funktionen

$$f(x) = 3 + e^{-|x|/10} - 2e^{x^2 e^{-|x|}}$$

som bestemmes af følgende underprogram.

Indtastning

g P/R
f LBL ,0
g ABS
CHS
 e^x
 $x \geq y$
g x^2
 $\times$
 e^x
2
 $\times$
CHS
 $x \geq y$
g ABS
CHS
1
0
 $\div$
 e^x

Lyspanel

000 –
001 – 42.21 ,0
002 – 43 16
003 – 16
004 – 12
005 – 34
006 – 43 11
007 – 20
008 – 12
009 – 2
010 – 20
011 – 16
012 – 34
013 – 43 16
014 – 16
015 – 1
016 – 0
017 – 10
018 – 12

Programmeringsstatus.

$e^{-|x|}$.

Lagrer x i X.

$x^2 e^{-|x|}$.

$-2e^{x^2 e^{-|x|}}$.

Lagrer x i X.

$-|x|/10$.

Indtastning

+
 3
+
g RTN

Lyspanel

019 – 40
020 – 3
021 – 40
022 – 43 32

$$e^{-|x|/10} - 2e^{x^2e^{-|x|}}$$

$$3 + e^{-|x|/10} - 2e^{x^2e^{-|x|}}$$

Brug SOLVE med følgende *enkelt-startværdier*: 10, 1 og 10^{-20} .

Indtastning

g P/R
 10 ENTER
f SOLVE .0
←
R↓
R↓
g R↑ g R↑
f SOLVE .0
←
R↓ R↓

Lyspanel

10,0000
Error 8
455,4335
48.026.721,85
1,0000
455,4335
Error 8
48.026.721,85
1,0000

Normalstatus.
 Enkelt startværdi.
 Bedste x -værdi.
 Foregående værdi.
 Funktionsværdi.
 Stakken retableres.
 Ny x -værdi.
 Samme funktionsværdi
 (asymptote).
 Enkelt startværdi.

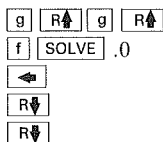
1 ENTER
f SOLVE .0
←
R↓
R↓
g R↑ g R↑
f SOLVE .0
←
R↓ R↓

1,0000
Error 8
2,1213
2,1471
0,3788
2,1213
Error 8
2,1213
0,3788

Bedste x -værdi.
 Foregående værdi.
 Funktionsværdi.
 Retablering af stakken.

EEX CHS 20 ENTER **1,0000 – 20**
f SOLVE .0 **Error 8**
← **1,0000 – 20**
R↓ **1,1250 – 20**
R↓ **2,0000**

Samme x -værdi.
 Samme funktionsværdi
 (asymptote).
 Enkelt startværdi.
 Bedste x -værdi.
 Foregående værdi.
 Funktionsværdi.

Indtastning**Lyspanel**

1,0000 **- 20**
Error 8
1,1250 **- 20**
1,5626 **- 16**
2,0000

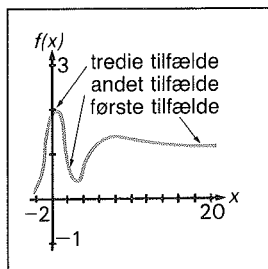
Retablering af stakken.

Ny x -værdi.

Foregående værdi.

Samme funktionsværdi.

I hvert af disse tre tilfælde søgte SOLVE først efter en rod i den retning, som antydedes af startværdien. Ved at bruge 10 som startværdi fandt SOLVE den vandrette asymptote (med værdien 1,0000). Med 1 som startværdi fandt den et lokalt minimum på 0,3788 for $x = 2,1213$. Med 10^{-20} som startværdi var funktionen for alle praktiske formål konstant lig 2,0000 inden for det lille x -interval, der blev brugt.



Ligninger med flere løsninger

Mange af de ligninger De møder vil have mere end én rod. Af denne grund vil De ofte have glæde af at forstå teknikken bag nogle af de metoder, der kan bruges til at finde flere rødder.

Den enkleste metode til at finde flere løsninger er at styre rodsøgningen ind mod forskellige intervaller, hvor der er stor sandsynlighed (eller sikkerhed) for at finde rødder. Startværdierne angiver det interval, som først bliver efterprøvet. Sådan har vi gjort i eksemplerne i afsnit 13 og De vil, som nævnt, ofte kunne finde løsninger på denne måde.

En anden metode er kendt som "reduktion". Når De har fundet en løsning, reduceres den oprindelige ligning med hensyn til den fundne rod. I dette tilfælde skal De altså omformulere ligningen efter hver ny rod.

Hvis en funktion $f(x)$ har et nulpunkt for $x = a$ vil funktionen $f(x)/(x - a)$ ikke have samme nulpunkt (det forudsættes at a er en enkeltrod). De kan bruge denne information til at eliminere en kendt rod. De skal blot indstille et underprogram på nogle få linier i slutningen af funktionsrutinen. Disse linier skal subtrahere den kendte rod (som skal opgives med 10

betydende cifre) fra x -værdien og derefter dividere funktionsværdien med differencen. I de fleste tilfælde er der kun én rod med samme værdi, og den nye funktion vil føre SOLVE væk fra den kendte rod.

På den anden side er det muligt at der er fundet en *multipl* rod; dvs. en rod, som synes at fremkomme gentagne gange i samme punkt. Dette sker, hvis $f(x)$ (og måske de følgende afledte) er nul for $x = a$. Det betyder så, at hvis roden er multipl, kan De ikke eliminere den ved en simpel division. De kan f.eks. betragte funktionen

$$f(x) = x(x - a)^3 = 0$$

som har en multipl rod for $x = a$ (tredobbelt rod). Denne rod kan ikke elimineres ved division med $(x - a)$. Men den kan elimineres med $(x - a)^3$.

Eksempel: Brug reduktion til at finde rødderne i

$$60x^4 - 944x^3 + 3003x^2 + 6171x - 2890 = 0.$$

Brug Horner's metode til at omskrive polynomiet til

$$(((60x - 944)x + 3003)x + 6171)x - 2890 = 0.$$

De skal nu skrive et program, der kan beregne polynomiet.

Indtastning

g	P/R
f	CLEAR PRGM
f	LBL 2
6	
0	
x	
9	
4	
4	
-	
x	
3	
0	

Lyspanel

000 -	
000 -	
001 -	42.21. 2
002 -	6
003 -	0
004 -	20
005 -	9
006 -	4
007 -	4
008 -	30
009 -	20
010 -	3
011 -	0

Programmeringsstatus.

Indtastning	Lyspanel
0	012 – 0
3	013 – 3
+	014 – 40
×	015 – 20
6	016 – 6
1	017 – 1
7	018 – 7
1	019 – 1
+	020 – 40
×	021 – 20
2	022 – 2
8	023 – 8
9	024 – 9
0	025 – 0
–	026 – 30
g RTN	027 – 43 32

I normalstatus skal De nu indtaste to store, negative startværdier (som f.eks. -10 og -20) og bruge SOLVE til at finde den numerisk største, negative rod.

Indtastning	Lyspanel	
g P/R		Normalstatus.
10 CHS ENTER	– 10,0000	} Startværdier.
20 CHS	– 20	
f SOLVE 2	– 1,6667	Første rod.
STO 0	– 1,6667	Lagring af første rod (til reduktion).
R↕ R↕	4,0000 – 06	Funktionsværdi nær nul-punkt.

Indtastning	Lyspanel	
g P/R	000 –	Programmeringsstatus.
g BST g BST	026 – 30	Linien før RTN .
$x \geq y$	027 – 34	Lagrer x i X.
RCL 0	028 – 45 0	} Dividerer med $(x - a)$, hvor a er en kendt rod.
–	029 – 30	
÷	030 – 10	

Brug nu samme startværdier til at finde næste rod.

Indtastning	Lyspanel	
g P/R	4,0000 - 06	Normalstatus.
10 CHS ENTER	- 10,0000	} Samme startværdier.
20 CHS	- 20	
f SOLVE 2	0,4000	Anden rod.
STO 1	0,4000	Lagring af anden rod (til reduktion).
R↕ R↕	0,0000	Ny funktionsværdi.

De skal nu modificere underprogrammet, så det kan eliminere den anden rod.

Indtastning	Lyspanel	
g P/R	000 -	Programmeringsstatus.
g BST g BST	030 - 10	Linien før RTN .
x≥y	031 - 34	Lagrer x i X.
RCL 1	032 - 45 1	} Dividerer med $(x - a)$, hvor a er den nye rod.
-	033 - 30	
÷	034 - 10	

Brug nu samme startværdier som før:

Indtastning	Lyspanel	
g P/R	0,000	Normalstatus.
10 CHS ENTER	- 10,0000	} Samme startværdier.
20 CHS	- 20	
f SOLVE 2	8,4999	Tredie rod.
STO 2	8,4999	Lagring af tredie rod (til reduktion).
R↕ R↕	- 1,0929 - 07	Ny funktionsværdi.

De skal nu modificere underprogrammet, så det kan eliminere den tredie rod.

Indtastning

g P/R
g BST g BST
 $x \geq y$
RCL 2
-
÷

Lyspanel

000 –
034 – **10**
035 – **34**
036 – **45 2**
037 – **30**
038 – **10**

Programmeringsstatus.

Linien før RTN.Lagrer x i X.
 Dividerer med $(x - a)$, hvor a
 er den nye rod.

Brug nu samme startværdier som før:

Indtastning

g P/R
 10 CHS ENTER
 20 CHS
f SOLVE 2
STO 3
R↓ R↓

Lyspanel

- 1,0929 - 07
- 10,0000
- 20
8,5001
8,5001
- 0,0009

Normalstatus.

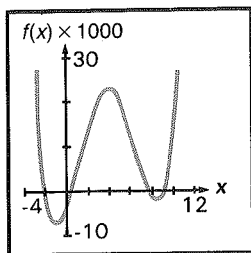
Samme startværdier.

Fjerde rod.

 Lagring af fjerde rod (til re-
 ferencebrug).

Ny funktionsværdi.

Ved at bruge samme sæt startværdier hver gang har De nu fundet dette fjerdegradspolynomiums fire rødder. Men de to sidst fundne rødder ligger meget tæt, så De har anledning til at forvente, at der er tale om en dobbeltrod. Det viser sig også, at De ikke uden videre kan eliminere den tredje rod. Der er faktisk tale om en dobbeltrod: for $x = 8,5$ bliver $f(x)$ eksakt lig nul, og værdierne 8,4999 og 8,5001 fremkommer alene på grund af afrundingsfejl.

**Graf for $f(x)$**

I almindelighed vil De ikke kende multipliciteten af den rod, De prøver at eliminere. Hvis De, efter forsøg på at eliminere en rod, får samme rod beregnet af SOLVE, kan De fortsætte på flere måder. De kan:

- Bruge nye startværdier på det reducerede udtryk i et forsøg på at søge efter en anden rod.
- Reducere udtrykket igen således, at den multiple rod elimineres. Hvis der ikke blot er tale om en dobbeltrod, vil De skulle gentage denne proces.
- Undersøge den reducerede funktions adfærd for x -værdier nær den kendte rod. Hvis de beregnede funktionsværdier skærer y -aksen uden spring, er der enten tale om endnu en rod eller om en multipel rod.
- Analysere den originale funktion og dens afledte algebraisk. Det kan herved blive muligt at bestemme funktionens adfærd nær et kendt nulpunkt. (F.eks. kan en Taylor-rækkeudvikling ofte bruges til at bestemme en multiplicitet.)

Begrænsning af beregningstiden

I nogle tilfælde vil De finde det ønskeligt at kunne begrænse den tid, der bruges af SOLVE til at finde en rod. Det kan gøres på to principielt forskellige måder: ved begrænsning af antallet af iterationer og ved begrænsning af nøjagtighed.

Begrænsning af antal iterationer

Når SOLVE søger efter en rod, vil den typisk beregne mindst 10 funktionsværdier. Men i nogle enkelte tilfælde vil SOLVE udføre flere hundrede beregninger. (Men SOLVE vil dog altid stoppe af sig selv.) Da underprogrammet, som beregner funktionen, bliver brugt for hvert nyt estimat, kan underprogrammet selv bruges til at tælle og begrænse antallet af iterationer. Det kan f.eks. gøres ganske let med ISG, som kan akkumulere antallet af iterationer i indeksregistret (eller et vilkårligt andet dataregister).

Hvis De lagrer en sådan tæller i et dataregister, vil SOLVE senest stoppe, når det tilladte antal iterationer er udført.

Begrænsning af nøjagtighed

De kan begrænse søgetiden ved at angive en største acceptabel unøjagtighed for funktionen. Det betyder, at underprogrammet skal sætte funktionsværdien til nul, hvis den beregnede funktionsværdi numerisk ligger inden for et givet interval (tolerancen). Tolerancen skal svare til en værdi, som må anses for negligabel i praktisk arbejde, eller til en værdi, der svarer til beregningens nøjagtighed. Denne metode giver ikke blot en betragtelig tidsgevinst, den vil ofte give et sandere billede af roden. Specielt fritager den Dem for at vente unødigt længe på en unødvendig præcision. (Eksemplet side 222 udnytter denne teknik.)

Reference til yderligere information

I *HP-15C Advanced Functions Handbook* kan De finde yderligere information om teknik bag og anvendelser af SOLVE. Blandt de emner, der behandles, er:

- Brug af SOLVE i forbindelse med polynomier.
- Løsning af ligningssystemer.
- Lokalisering af lokale ekstremumpunkter.
- Brug af SOLVE til finansielle funktioner.
- Brug af SOLVE i kompleksstatus.
- Løsning af ligninger med komplekse rødder.

Tillæg E

Detaljeret gennemgang af $\int_y^x$

I afsnit 14, Numerisk integration, blev grundlaget for brug af $\int_y^x$ gennemgået. I dette tillæg diskuterer vi de mere dybtgående aspekter af $\int_y^x$, som vil interessere Dem, hvis De ofte skal arbejde med den numeriske integration.

Mekanikken bag $\int_y^x$

$\int_y^x$ -algoritmen beregner integralet af en funktion $f(x)$ ved at beregne den vægtede middelværdi af funktionsværdierne for en række værdier af x inden for det opgivne integrationsinterval. Den nøjagtighed, man kan opnå ved en sådan intervalopdeling, afhænger i første række af antallet af delintervaller: generelt gælder, at jo flere delintervaller, desto bedre nøjagtighed. Hvis $f(x)$ kunne beregnes i et uendeligt antal punkter over intervallet, ville integralet af $f(x)$ kunne beregnes eksakt – hvis ikke det endelige antal beregningscifre, præcisionen, satte en grænse for den opnåelige nøjagtighed.

Det ville tage meget lang tid at beregne funktionsværdien i et uendeligt antal punkter (man ville aldrig blive færdig). Men det er nu heller ikke nødvendigt for at opnå det optimale resultat. Den højest opnåelige nøjagtighed er begrænset af nøjagtigheden af de enkelte funktionsværdier. Når man arbejder med et endeligt antal delintervaller kan man således opnå et resultat, der er netop så nøjagtigt som man kan forvente, når den givne unøjagtighed i funktionen $f(x)$ tages i betragtning.

$\int_y^x$ opdeler først hovedintervallet i nogle ganske få delintervaller, således at der opnås relativt unøjagtige tilnærmelser. Hvis disse tilnærmelser ikke er så nøjagtige, som beregningen af $f(x)$ egentlig giver mulighed for, vil algoritmen blive gentaget for et større antal delintervaller. For hvert nyt sæt iterationer fordobles antallet af delintervaller. Denne proces fortsættes, indtil integralet er beregnet med en nøjagtighed, der svarer til funktionens nøjagtighed.

Usikkerheden på den endelige tilnærmelse er et tal, som afledes af udlæsningsformatet, som angiver usikkerheden på *funktionen*.^{*} Efter afslutningen af hvert sæt iterationer vil algoritmen sammenligne den nyeste tilnærmelse med de to foregående. Hvis differensen mellem to af disse tre tilnærmelser er mindre end den acceptable unøjagtighed stopper algoritmen, hvorved den nyeste tilnærmelse lagres i X og den tilhørende usikkerhed i Y .

Det er overordentlig usandsynligt, at fejlene i hver af disse tre successive tilnærmelser – dvs. differenserne mellem det faktiske integral og tilnærmelserne – er større end de indbyrdes forskelle mellem tilnærmelserne. Derfor vil fejlen på den endelige tilnærmelse være mindre end dens usikkerhed.^{**} Selv om vi af naturlige grunde ikke kan kende fejlen på den endelige tilnærmelse, er den med meget stor sandsynlighed mindre end den udlæste usikkerhed, som således udgør en næsten sikker øvre grænse for differensen mellem integralets faktiske værdi og tilnærmelsen.

Nøjagtighed, usikkerhed og beregningstid

Det er ikke på forhånd sikkert, at nøjagtigheden forbedres, hvis De forøger funktionsnøjagtigheden med et enkelt betydende ciffer. Men usikkerheden vil dog falde. På samme måde vil der være tilfælde, hvor beregningstiden ikke forlænges, når præcisionen skærpes.

Eksempel: Bessels funktion af fjerde orden kan udtrykkes således:

$$J_4(x) = \frac{1}{\pi} \int_0^\pi \cos(4\theta - x \sin \theta) d\theta.$$

^{*} Forholdet mellem udlæsningsformatet, funktionens usikkerhed og usikkerheden på integraltilnærmelsen bliver diskuteret senere i dette tillæg.

^{**} Under forudsætning af, at $f(x)$ ikke varierer meget hurtigt – et forhold, som vi vil komme tilbage til.

Beregn integralet i udtrykket for $J_4(1)$,

$$\int_0^{\pi} \cos(4\theta - \sin\theta) d\theta.$$

Skift først til programmeringsstatus og indtast et underprogram til beregning af funktionen $f(\theta) = \cos(4\theta - \sin\theta)$.

Indtastning

$\boxed{g}$ $\boxed{P/R}$
 $\boxed{f}$ $\boxed{CLEAR}$ $\boxed{PRGM}$
 $\boxed{f}$ $\boxed{LBL}$ 0
 4
 $\boxed{\times}$
 $\boxed{x \geq y}$
 $\boxed{SIN}$
 $\boxed{-}$
 $\boxed{COS}$
 $\boxed{g}$ $\boxed{RTN}$

Lyspanel

000 –
 000 –
 001 – 42.21. 0
 002 – 4
 003 – 20
 004 – 34
 005 – 23
 006 – 30
 007 – 24
 008 – 43 32

Programmeringsstatus.

Skift nu til normalstatus og indtast integrationsgrænserne i X og Y. Sørg for at vinkelmålet er sat til radian og at udlæsningsformatet er $\boxed{SCI}$ 2. Tryk $\boxed{f}$ $\boxed{\int y}$ 0 for at beregne integralet.

Indtastning

$\boxed{g}$ $\boxed{P/R}$
 0 $\boxed{ENTER}$
 $\boxed{g}$ $\boxed{\pi}$
 $\boxed{g}$ $\boxed{RAD}$
 $\boxed{f}$ $\boxed{SCI}$ 2

 $\boxed{f}$ $\boxed{\int y}$ 0
 $\boxed{x \geq y}$

Lyspanel

0,0000
 3,1416
 3,1416
 3,14 00

 7,79 – 03
 1,45 – 03

Normalstatus.

Indtast nedre grænse.

Indtast øvre grænse.

Vælg radian som vinkelmål.

Sæt udlæsningsformatet til $\boxed{SCI}$ 2.

Tilnærmelse med $\boxed{SCI}$ 2.

Usikkerhed med $\boxed{SCI}$ 2.

Usikkerheden antyder, at ikke alle nøjagtige cifre i tilnærmelsen bliver udlæst med $\boxed{SCI}$ 2. Det er faktisk således, at tilnærmelsen er langt mere nøjagtig end usikkerheden lader forstå.

Indtastning	Lyspanel	
$x \approx y$	7,79 - 03	Tilnærmelsen udlæses igen.
$\int$ CLEAR PREFIX		
(hold)	7785820888	Alle 10 cifre i tilnærmelsen bliver udlæst.

Den faktiske værdi af integralet er, med 5 cifres nøjagtighed, $7,7805 \times 10^{-3}$. Fejlen på tilnærmelsen er derfor ca. $(7,7858 - 7,7805) \times 10^{-3} = 5,3 \times 10^{-6}$. Denne fejl er væsentlig mindre end usikkerheden, $1,45 \times 10^{-3}$. Her har vi altså et eksempel på, at usikkerheden alene er en øvre grænse for tilnærmelsens fejl; den faktiske fejl vil normalt være mindre.

Bestem nu integralet med $\int$ 3 og sammenlign resultatet med tilnærmelsen, der blev opnået med $\int$ 2.

Indtastning	Lyspanel	
$\int$ $\int$ 3	7,786 - 03	Udlæsningsformatet er nu $\int$ 3.
$\downarrow$ $\downarrow$	3,142 00	Rul ned indtil den øvre grænse bliver udlæst.
$\int$ $\int$ 0	7,786 - 03	Integraltilnærmelse med $\int$ 3.
$x \approx y$	1,448 - 04	Usikkerhed med $\int$ 3.
$x \approx y$	7,786 - 03	Tilnærmelse udlæses atter.
$\int$ CLEAR PREFIX		
(hold)	7785820888	Alle 10 cifre.

Tilnærmelserne med $\int$ 2 og $\int$ 3 er identiske til 10 cifre: nøjagtigheden af tilnærmelsen med $\int$ 3 er ikke bedre end med $\int$ 2, selv om usikkerheden er mindre med $\int$ 3 end med $\int$ 2. For at forstå dette må man tænke på, at nøjagtigheden af en tilnærmelse først og fremmest afhænger af antallet af delintervaller. $\int$ $\int$ -algoritmen gentages med et stadigt voksende antal delintervaller indtil forskellen mellem tre på hinanden følgende tilnærmelser er mindre end den usikkerhed, der følger fra udlæsningsformatet. Efter en bestemt iteration kan forskellen være så meget mindre end usikkerheden, at den stadig ville være mindre, hvis usikkerheden blev en faktor 10 mindre. I sådanne tilfælde bliver integrationsintervallet ikke yderligere opdelt ved at De specificerer et større

antal cifre i udlæsningsformatet. Det betyder at tilnærmelsen bliver den samme som beregnet med den større usikkerhed.

Hvis De beregnede de to foregående tilnærmelser på Deres lommeregner, bemærkede De måske, at det ikke tog længere tid at beregne tilnærmelsen med SCI 3 end med SCI 2. Det er naturligvis fordi beregningstiden – alle andre forhold lige – alene afhænger af antallet af delintervaller. Som netop set, blev integrationsintervallet ikke yderligere opdelt ved skiftet fra SCI 2 til SCI 3. Derfor tog beregningen ikke længere tid.

Men i mange tilfælde vil det kræve flere delintervaller, hvis man ønsker at forøge antallet af betydende cifre. Lad os betragte forløbet med SCI 4.

Indtastning

Lyspanel

f SCI 4
R R

7,7858 – 03
3,1416 00

Skift til SCI 4.

Rul ned til den øvre grænse bliver udlæst.

f A 0

7,7807 – 03

Integraltilnærmelse med SCI 4.

Denne tilnærmelse tog ca. dobbelt så lang tid at udføre som tilnærmelsen med SCI 2 eller SCI 3. I dette tilfælde skulle funktionsværdien for funktionen beregnes dobbelt så ofte som før for at opnå en tilnærmelse af acceptabel nøjagtighed. Bemærk dog at De bliver belønnet for Deres tålmodighed: denne tilnærmelse er væsentligt nøjagtigere end den foregående – fejlen er nu 2 på 5. cifre, hvor den før var 6 på 4. cifre; De får således næsten 2 cifre bedre nøjagtighed.

De foregående eksempler viser, at De kun somme tider får en nøjagtighedsgevinst ved at vælge en udlæsning med et større antal cifre. Hvorvidt det ene eller andet sker, afhænger ganske af den enkelte funktions egenskaber inden for integrationsintervallet og kan normalt kun afgøres ved at De prøver.

Hvis De opnår en nøjagtighedsgevinst sker det på bekostning af hastigheden: programudførelsen vil stige til næsten det dobbelte for hvert nyt udlæst ciffer. Der er ingen måde, hvorpå De kan undgå dette. Nøjagtighed og antal delintervaller hænger sammen og antal delintervaller og tidsforbrug hænger sammen.

Den tid, der går med at beregne et integral, er ikke alene afhængig af antallet af udlæste cifre, men også i nogen grad af intervallets grænser. Hvis det tager uforholdsmæssigt lang tid at beregne et integral kan det skyldes, at intervallet er for bredt i forhold til funktionens adfærd i intervallet. I de fleste tilfælde behøver De dog ikke bekymre Dem om grænsernes indflydelse på beregningstiden. De specielle tilfælde, der kan påvirke beregningstiden, vil blive omtalt senere i dette tillæg.

Usikkerhed og udlæsningsformat

På grund af afrundingsfejl kan underprogrammet, der bruges til at beregne $f(x)$, ikke beregne funktionsværdierne helt eksakt, men beregner i stedet

$$\hat{f}(x) = f(x) \pm \delta_1(x),$$

hvor $\delta_1(x)$ er usikkerheden på $f(x)$ som følge af afrundingsfejl. Hvis $f(x)$ er et udtryk for et fysisk fænomen, vil funktionen De ønsker at integrere normalt snarere være

$$F(x) = f(x) \pm \delta_2(x),$$

end $f(x)$, hvor $\delta_2(x)$ er usikkerheden på $f(x)$ som følge af måleusikkerhed eller anden uundgåelig usikkerhed på den fysiske størrelse.

Da $f(x) = \hat{f}(x) \pm \delta_1(x)$ vil funktionen, der skal integreres, være:

$$F(x) = \hat{f}(x) \pm \delta_1(x) \pm \delta_2(x)$$

eller

$$F(x) = \hat{f}(x) \pm \delta(x),$$

hvor $\delta(x)$ er den samlede usikkerhed på $\hat{f}(x)$.

Integralet, der skal beregnes, er således:

$$\begin{aligned}\int_a^b F(x) dx &= \int_a^b [\hat{f}(x) \pm \delta(x)] dx \\ &= \int_a^b \hat{f}(x) dx \pm \int_a^b \delta(x) dx \\ &= I \pm \Delta\end{aligned}$$

hvor I er en tilnærmelse til $\int_a^b F(x) dx$ og Δ er usikkerheden på tilnærmelsen. $\boxed{f}$ -algoritmen lagrer I i X og Δ i Y .

Usikkerheden $\delta(x)$ på $\hat{f}(x)$, funktionen som beregnes af underprogrammet, bestemmes således: Antag at De anser funktionens tre første betydende cifre for nøjagtige, således at De vælger $\boxed{SCI}$ 2. Lyspanelet vil da kun udlæse de af funktionsværdiens cifre, som er nøjagtige. F.eks.

1,23 - 04

Da udlæsningsformatet afrunder tallet i X -registret til det udlæste tal, medfører dette, at usikkerheden på funktionsværdien er $\pm 0,005 \times 10^{-4} = \pm 0,5 \times 10^{-2} \times 10^{-4} = \pm 0,5 \times 10^{-6}$.

Ved at sætte udlæsningsformatet til $\boxed{SCI}$ n eller $\boxed{ENG}$ n , hvor n er et heltal,* bliver usikkerheden på funktionsværdien altså

$$\begin{aligned}\delta(x) &= 0.5 \times 10^{-n} \times 10^{m(x)} \\ &= 0.5 \times 10^{-n+m(x)}\end{aligned}$$

* Selv om $\boxed{SCI}$ 8 eller 9 i almindelighed resulterer i samme udlæsning som $\boxed{SCI}$ 7, vil disse udlæsningsformater medføre en mindre usikkerhed på det beregnede integral. (Det samme gælder for $\boxed{ENG}$.) En negativ værdi for n (som kan sættes med indeksregistret) vil påvirke usikkerheden i en $\boxed{f}$ -beregning. Den mindste værdi for n , som vil have nogen indflydelse på usikkerheden, er -6 . For n mindre end -6 sættes indholdet af R_1 til -6 .

I denne formel er n antallet af cifre, der er specificeret i udlæsningsformatet, og $m(x)$ er eksponenten for funktionsværdien i x , når $\boxed{\text{SCI}}$ bruges.

Usikkerheden er proportional med faktoren $10^{m(x)}$, som repræsenterer størrelsesordenen af funktionsværdien i x . Derfor vil $\boxed{\text{SCI}}$ og $\boxed{\text{ENG}}$ medføre en usikkerhed, der er *relativ* til funktionens størrelsesorden.

På samme måde gælder, at hvis funktionsværdien er udlæst med $\boxed{\text{FIX}}$ n , vil afrundingen af det udlæste tal medføre, at usikkerheden på funktionsværdien bliver

$$\delta(x) = 0.5 \times 10^{-n}.$$

Da denne usikkerhed er uafhængig af funktionsværdiens størrelse, vil $\boxed{\text{FIX}}$ medføre en *absolut* usikkerhed.

Hver gang $\int$ beregner funktionsværdien for x udleder den samtidig en værdi for $\delta(x)$, funktionens usikkerhed i x . Beregningen udføres med det antal cifre, n , der er valgt til udlæsningsformatet og (hvis der arbejdes med $\boxed{\text{SCI}}$ eller $\boxed{\text{ENG}}$) størrelsesordenen $m(x)$ af funktionsværdien i x .

Tallet Δ , usikkerheden på tilnærmelsen af det ønskede integral, er selv integralet af $\delta(x)$:

$$\begin{aligned}\Delta &= \int_a^b \delta(x) dx \\ &= \int_a^b [0.5 \times 10^{-n+m(x)}] dx.\end{aligned}$$

Dette integral beregnes i delintervallernes endepunkter på stort set samme måde som tilnærmelsen til integralet af funktionen $f(x)$.

Da Δ er proportional med faktoren 10^{-n} vil usikkerheden på en tilnærmelse ændres med en faktor 10 for hvert nyt ciffer i udlæsningsformatet. Dette vil dog ikke passe helt for $\boxed{\text{SCI}}$ - og $\boxed{\text{ENG}}$ -formaterne, da et skift i antallet af cifre ofte medfører, at funktionen skal beregnes i andre punkter, således at $\delta(x) \sim 10^{m(x)}$ får en anden værdi.

Bemærk, at når et integral tilnærmes med $\boxed{\text{FIX}}$ vil $m(x)$ være lig 0, således at den beregnede usikkerhed på tilnærmelsen bliver

$$\Delta = 0.5 \times 10^{-n}(b-a).$$

Normalt er det ikke nødvendigt at angive usikkerheden på funktionen særlig præcist. (Det kræver i de fleste tilfælde en meget omfattende analyse.) I almindelighed er det mest bekvemt at anvende SCI eller ENG , hvis usikkerheden på funktionsværdien lettest kan bestemmes relativt. Tilsvarende bør man anvende FIX , hvis det er lettest at bestemme den absolutte usikkerhed. FIX kan være uhensigtsmæssig (og lede til besynderlige resultater), hvis De integrerer en funktion, hvis størrelse og usikkerhed er ekstremt små inden for integrationsintervallet. På samme måde vil SCI kunne føre til barokke resultater, hvis funktionsværdierne bliver meget mindre end usikkerheden.

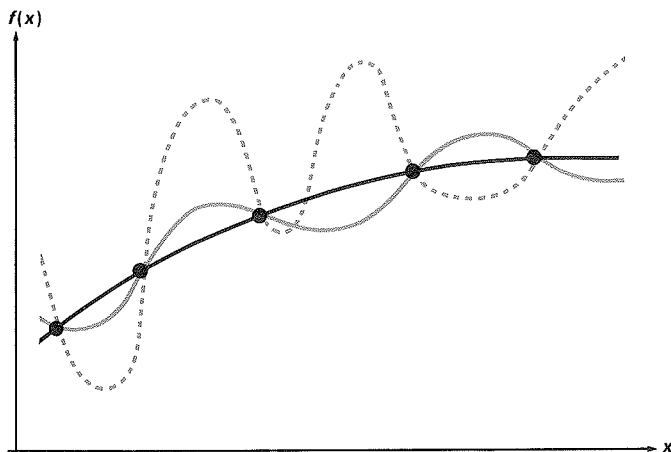
Hvis resultaterne af en integralberegning synes underlige, bør De forsøge at gennemføre beregningen med et radikalt andet udlæsningsformat.

Forhold der kan føre til forkerte resultater

Selv om β -algoritmen i HP-15C er blandt de bedste der fås, vil den i visse, specielle tilfælde gøre som snart sagt alle andre numeriske metoder: give et helt forkert resultat – uden nogen form for varsel. *Men sandsynligheden for at dette sker med HP-15C er overordentlig ringe.* β -algoritmen er udviklet så den giver nøjagtige resultater for næsten enhver glat funktion. Kun i forbindelse med funktioner, som udviser en ekstremt besynderlig adfærd, er der en anelse af fare for at få et fejlagtigt resultat. Sådanne funktioner forekommer kun sjældent i det virkelige liv – og når det endelig sker, er det som oftest muligt på anden måde at erkende funktionens egenart. I de fleste sådanne tilfælde kan problemet omgås ganske enkelt.

Som omtalt side 237 vil β beregne funktionsværdier for forskellige x over hele integrationsintervallet. Ved at bestemme den vægtede middelværdi af de beregnede funktionsværdier, $f(x)$, tilnærmer algoritmen integralet af $f(x)$.

Nu er det uheldigvis således, at algoritmen kun kender funktionen i de punkter, hvor den har beregnet funktionsværdier, og at den ikke har nogen mulighed for at skelne mellem forskellige funktioner, som har disse punkter tilfælles. En situation, hvor forskellige funktioner er beregnet i samme punkter, er illustreret side 246. De tre funktioner repræsenterer de uendeligt mange funktioner, hvis grafer omfatter de endeligt mange beregningspunkter.



Med dette antal beregningspunkter (svarende til et givet antal delintervaller) vil algoritmen beregne samme værdi for alle tre funktioners integraler. Integralerne for funktionerne, hvis grafer er vist med fuldt optrukne linier, er stort set lige store, så tilnærmelsen vil være rimeligt nøjagtig hvis $f(x)$ er en af disse funktioner. Men hvis $f(x)$ er funktionen, hvis graf er vist med en stiplede linie, tegner situationen sig ganske anderledes: tilnærmelsen vil være ret unøjagtig.

$\int$ lærer funktionen at kende ved at beregne funktionsværdier i flere og flere punkter. Hvis funktionens fluktuationer i ét område af integrationsintervallet ikke er ulig fluktuationsmønstret over resten af intervallet er det yderst tænkeligt, at algoritmen bemærker fluktuationen. Når dette sker, vil antallet af delintervaller blive forøget indtil successive iterationer tager højde for tilstedeværelsen af de hurtigste, *men karakteristiske*, fluktuationer.

Betragt f.eks. tilnærmelsen til

$$\int_0^{\infty} x e^{-x} dx.$$

Da De bestemmer dette integral numerisk, synes De måske (lidt naivt, som vi straks skal se) at De bør sætte den øvre integrationsgrænse til 10^{99} , som jo er noget nær det største tal, som kan repræsenteres af HP-15C. Prøv nu, og se hvad der sker.

Indtast et underprogram, som bestemmer funktionen $f(x) = xe^{-x}$.

Indtastning

g P/R
f LBL 1
CHS
 e^x
×
g RTN

Lyspanel

000 –
 001 – 42.21. 1
 002 – 16
 003 – 12
 004 – 20
 005 – 43 32

Programmeringsstatus.

Skift til normalstatus. Sæt udlæsningsformatet til SCI 3 og indtast integrationsgrænserne i X og Y.

Indtastning

g P/R
f SCI 3
 0 ENTER
EEX 99
f $\int$ 1

Lyspanel

0,000 00
 1 99
 0,000 00

Normalstatus.

Sætter udlæsningsformatet til SCI 3.

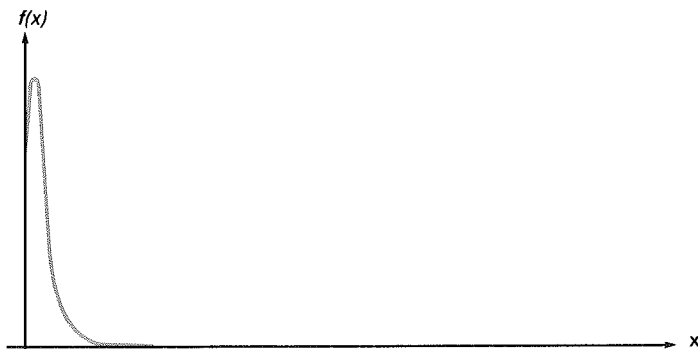
Indtast nedre grænse.

Indtast øvre grænse.

Tilnærmelse.

Dette svar er tydeligvis forkert – værdien af integralet skal for det specificerede interval være eksakt 1. Men problemet er *ikke*, at De repræsenterede ∞ med 10^{99} , da værdien er integralet for dette interval rigeligt er så nær 1, som HP-15C kan komme.

Problemet bliver klart, når man betragter funktionens graf over integrationsintervallet:

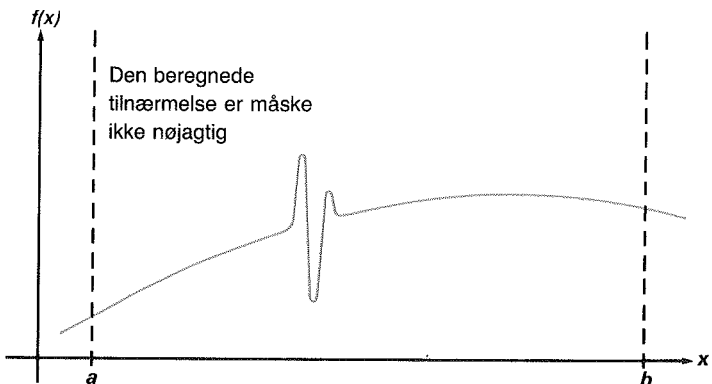
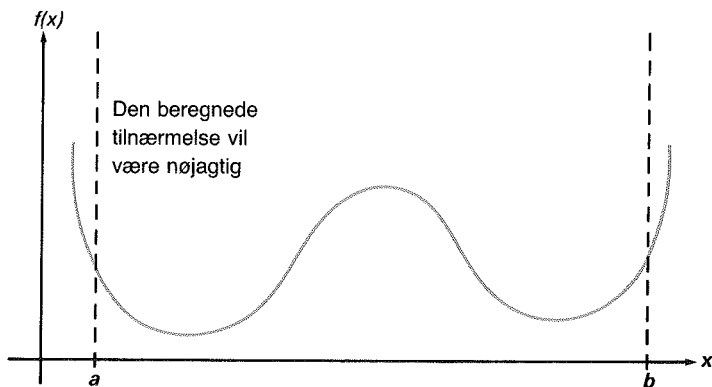


Grafen har en spids meget nær $x = 0$. (Spidsen er langt spidsere i virkeligheden end vist på tegningen. Hvis den skulle vises i samme målestok som resten af tegningen, ville den ikke kunne skelnes fra Y-aksen.) Ved intervalopdelingen er intet intervalendepunkt faldet ind under spidsen, så algoritmen har aldrig fundet andre funktionsværdier end 0. Selv hvis De prøvede at forøge antallet af delintervaller ved at skifte til $\boxed{\text{SCI}}$ 9, er intervallet om spidsen stadig så snævert, at det falder ind under et enkelt delinterval. (I slutningen af næste afsnit, Forhold der forlænger beregningstiden, er der omtalt metoder, som i visse tilfælde kan løse denne type problemer.)

De har nu set, hvorledes $\boxed{\int}$ kan give forkerte resultater, hvis der optræder en lokal, dvs. ukarakteristisk, fluktuation. Heldigvis er funktioner med så afvigende adfærd så sjældne, at det er yderst usandsynligt at De uafvidende begynder at anvende standardmetoder til integralberegningen.

Funktioner som kan lede til forkerte resultater kan identificeres ret let ved at beskrive hvor hurtigt den og dens første afledede varierer over integrationsintervallet. I det store og hele kan man sige, at desto mere og hurtigere en funktion og dens første afledede varierer, desto langsommere udføres beregningen og desto mindre nøjagtig bliver den.

Bemærk, at variationshastighederne skal bestemmes med hensyn til intervallets længde. En funktion hvis fluktuationer er spredt ud over hele integrationsintervallet kan bedre karakteriseres ved de gennemførte funktionsberegninger end en funktion, hvor en række fluktuationer er samlet inden for et lille interval. (Se f.eks. den følgende illustration.) Vi kan betragte variationerne eller fluktuationerne som en slags oscillation og interessere os for forholdet mellem oscillationens periode og integrationsintervallets længde. Jo større dette forhold er, desto hurtigere og mere pålideligt vil integralet blive tilnærmet.



I mange tilfælde vil De kende funktionen godt nok til at De kan afgøre, om den udviser en speciel adfærd inden for det interval, der skal arbejdes med. Hvis De ikke kender funktionen, men har mistanke om variationer, der er store i forhold til intervallængden, kan De ofte klare problemet ved at udføre nogle funktionsberegninger for forskellige værdier af x , således at De kan grovplotte funktionen.

Hvis De af en eller anden grund får mistanke til et beregnet resultat, kan De anvende en meget enkel metode til at vurdere dets pålidelighed. Inddel integrationsintervallet i et eller flere underintervaller, beregn integralet for hvert enkelt interval og summér de fundne værdier. Herved vil algoritmen blive tvunget til at beregne funktionen for helt nye værdier af x , og chancen for at finde hidtil upågtede spidser vokser således. Hvis den nye sum svarer til den oprindelige tilnærmelse, vil De normalt kunne tage dette som en bekræftelse af tilnærmelsen.

Forhold der forlænger beregningstiden

I det foregående eksempel (side 247) så De at algoritmen gav et forkert svar fordi den aldrig opdagede spidsen i funktionen. Det skyldtes, at variationen var for hurtig i forhold til intervallets længde. Hvis integrationsintervallet havde været mindre, ville sandsynligheden for at algoritmen fandt spidsen være større. Hvis intervallet var smalt nok, ville resultatet blive korrekt. Men beregningstiden ville være meget lang, hvis ikke intervallet var meget smalt.

For visse integraler, som f.eks. det netop omtalte, vil beregningstiden blive urimeligt lang på grund af at integrationsintervallet er for stort i forhold til visse egenskaber ved den funktion der integreres.

Betragt nu i stedet et integral, hvor integrationsintervallet er bredt nok til at kræve en meget lang beregningstid, men samtidig smalt nok til at give et korrekt resultat. Bemærk at da $f(x) = xe^{-x}$ går meget hurtigt mod nul for x gående mod uendelig, vil integrationsbidraget for store værdier af x være negligeable. Vi vil derfor nu bestemme integralet med en ny øvre grænse – f.eks. 10^3 .

Indtastning

0 ENTER

EEX 3

 f $\int$ 1

 $x \geq y$
Lyspanel

0,000 00

1 03

1,000 00

1,824 -04

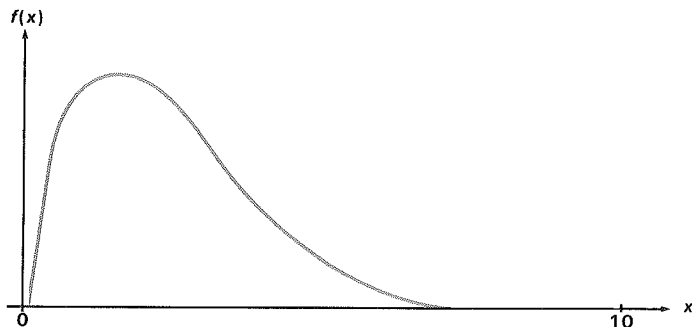
Indtast nedre grænse.

Indtast øvre grænse.

Tilnærmelse.

Usikkerhed på tilnærmelse.

Dette er det korrekte svar – men det tog meget lang tid at nå frem til det. For at forstå hvorfor kan De sammenligne grafen for funktionen over det netop anvendte integrationsinterval (den svarer ganske nøje til den, der blev vist side 248) med grafen for funktionen over intervallet 0 til 10.



Ved at sammenligne de to grafer kan De tydeligt se, at funktionen kun er "interessant" for små værdier af x . For større værdier af x bliver funktionen "uinteressant" fordi den aftager jævnt og gradvist på en overordentlig forudsigelig måde.

Som tidligere omtalt vil $\int$ -algoritmen inddеле integrationsintervallet i stadigt flere delintervaller indtil forskellen mellem successive iterationsresultater bliver tilstrækkeligt lille. Algoritmen vil med andre ord bestemme funktionsværdierne for et stigende antal x -værdier indtil den har tilstrækkelig information om funktionen til at kunne præstere en tilnærmelse, som kun ændres ubetydeligt ved indsamling af yderligere information.

Hvis vi bruger 0 og 10 som øvre og nedre integrationsgrænser, således at algoritmen udelukkende behøver at beregne funktionsværdier i et interval, der er interessant men relativt jævnt, vil nye funktionsberegninger efter ganske få iterationer ikke tilføre ny information om funktionen. Der skal derfor kun udføres ganske få iterationer før forskellen mellem successive forsøg bliver så ringe, at algoritmen kan afsluttes med en tilnærmelse af acceptabel nøjagtighed.

Hvis, på den anden side, integrationsintervallet snarere minder om det, der blev brugt på side 248, vil de fleste delintervaldepunkter ligge i et område, hvor funktionen ikke ændrer sig meget. De få beregninger for små x vil bemærke, at funktionsværdien skifter betragteligt fra den ene iteration til den næste. Funktionen skal derfor beregnes igen og igen med stadigt snævrere delintervaller før forskellen mellem successive iterationer bliver acceptabel.

Da beregningstiden i så høj grad afhænger af hvor hurtigt der opnås en vis tæthed af delintervaller i et område, hvor funktionen er interessant, vil beregningstiden for ethvert integral blive forlænget, hvis integrationsintervallet i alt væsentligt omfatter områder, hvor funktionen er uinteressant. Heldigvis vil De i de fleste tilfælde kunne modificere problemet således, at De opnår en betragtelig reduktion af beregningstiden. To sådanne metoder er intervalopdeling og variabeltransformation. Med disse metoder kan De ændre funktionen eller integrationsintervallet således, at integrandens adfærd over intervallet (eller intervallerne) er bedst mulig. (Disse metoder omtales nærmere i *HP-15C Advanced Functions Handbook*.)

Udlæsning af foreløbig tilnærmelse

Når en integralberegning tager længere tid end De har lyst til at vente, kan De afbryde algoritmen og få udlæst den seneste tilnærmelse. Men De kan ikke få udlæst usikkerheden på denne tilnærmelse.

Hvis De trykker på  medens HP-15C beregner et integral, vil beregningen stoppe på samme måde som under sædvanlig programkørsel. Programpilens pege på den programlinje, der var under udførelse, og lyspanelet udlæser resultatet af beregningen i den foregående programlinje. Bemærk, at når De afbryder beregningen, vil den aktuelle tilnærmelse *ikke* være lagret i X eller noget andet stakregister. Som ved ethvert andet program vil fornyet tryk på  genstarte programmet fra programpilens placering.

 opdaterer den aktuelle tilnærmelse og lagrer den i SIDSTE X efter hver ny funktionsberegning. De kan derfor få udlæst den aktuelle tilnærmelse ved simpelt hen at stoppe kørslen og bruge  til at færdiggøre funktionsunderprogrammet. Derefter får De udlæst tilnærmelsen med  , som blev opdateret da  blev udført.

Når den aktuelle tilnærmelse opdateres, slettes lyspanelet, og **running** bliver ikke udlæst, som når underprogrammet kører. Hvis De derfor kan stoppe lommeregneren medens lyspanelet er blankt, kan De undgå at skulle gennemgå underprogrammet linie for linie.

De kan få udlæst en foreløbig tilnærmelse af et integral ved at følge nedenstående procedure:

1. Tryk på , så programmet stopper – helst medens lyspanelet er blankt.
2. Når programmet stopper, skal De skifte til programmeringsstatus og se, hvilken programlinje programpilens peger på.
 - Hvis linien indeholder startetiketten for funktionsunderprogrammet, skal De skifte tilbage til normalstatus og udlæse indholdet af SIDSTE X (trin 3).
 - Hvis enhver anden programlinje udlæses, skal De skifte tilbage til normalstatus og bruge  indtil De når en  instruktion (tastkode 43 32) eller linie 000 (hvis der ikke er nogen ). (Sørg for at holde  så længe nedtrykket, at De får udlæst linienumre og tastkoder.)
3. Tryk  , så den foreløbige (aktuelle) tilnærmelse bliver udlæst. Hvis De ønsker at fortsætte beregningen fra dette punkt, skal De trykke   . Herved genfyldes stakken med den aktuelle x-værdi og beregningen genoptages.

Reference til yderligere information

HP-15C Advanced Functions Handbook gennemgår mere esoteriske aspekter af $\int$ og dens anvendelser. Det drejer sig blandt andet om:

- Nøjagtighed af integranden.
- Begrænsning af beregningstid.
- Beregning af vanskelige integraler.
- Brug af $\int$ i kompleksstatus.

Batteri, service og udbedring af mangler

Batterier

HP-15C drives af tre batterier. Maskinen leveres med 3 alkaliske knapceller, men også sølvoxidceller kan bruges.

Et sæt batterier bestående af 3 friske alkaliske knapceller holder til mindst 60 timers uafbrudt brug (den form for brug, der belaster batterierne mest). Hvis maskinen ikke bruges, vil batterierne alligevel ikke holde uendeligt; de udsættes for en vis selvafladning. Den kontinuerlige hukommelse påvirker ikke batterierne i nogen betydende grad; det betyder, at den kontinuerlige hukommelses indhold bevares indtil batteriernes klemspænding pludselig falder drastisk – det sker tidligst efter 1 år.

Et sæt friske sølvoxidbatterier holder til mindst 135 brugstimer. Sølvoxidbatterierne selvaflades lidt langsommere end de alkaliske celler, så man kan regne med, at den kontinuerlige hukommelse i hvert fald bevares i 2 år, hvis maskinen ikke bruges.

Den faktiske batterilevetid vil ligge et eller andet sted imellem disse yderpunkter og afhænger ganske af brugen: om maskinen bruges i længere perioder ad gangen, om den bruges manuelt eller til programkørsel.*

Bemærk, at batterierne *ikke* er genopladelige.

ADVARSEL

Batterierne kan eksplodere under voldsom trykudvikling, hvis De forsøger at genoplade batterierne, opbevarer dem nær en hed varmekilde eller kaster dem i åben ild.

* Levptiden afhænger også af brugstemperaturen: batterierne aflades meget hurtigt, hvis de bruges ved lave temperaturer, og selvafladningen foregår hurtigst ved høje temperaturer. De almindelige knapceller er beregnet til brug ved stuetemperatur.

De kan bruge følgende eller lignende batterier i HP-15C:

Alkaliske

IEC LR44
UCAR A76
Ray-O-Vac RW 82
BEREC LR44
Varta 4276
Toshiba LR44
Mallory 10L14

Sølvoxid

IEC SR44
UCAR 357
Ray-O-Vac RS 76 eller H-RW42
BEREC BSR44H eller BSR44L
Varta 541
Toshiba G13

Batteriindikator

En stjerne (★) der blinker i lyspanelets nederste venstre hjørne er tegn på, at batterierne skal udskiftes.

Ved brug af alkaliske batterier gælder:

- HP-15C kan bruges i mindst 1 time, efter at stjernen viser sig første gang.
- Hvis man slukker for maskinen, når stjernen begynder at blinke, kan den kontinuerlige hukommelses indhold bevares mindst 1 måned endnu.

Ved brug af sølvoxidbatterier gælder:

- HP-15C kan bruges i mindst 10 minutter, efter at stjernen viser sig første gang.
- Hvis man slukker for maskinen, når stjernen begynder at blinke, kan den kontinuerlige hukommelses indhold bevares mindst 1 uge endnu.

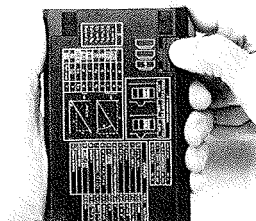
Udskiftning af batterier

Når De tager batterierne ud af HP-15C, vil den kontinuerlige hukommelses indhold blive bevaret så længe, at De skulle have rigelig tid til at sætte nye batterier på plads, forudsat at De slukker for maskinen inden batterierne tages ud.

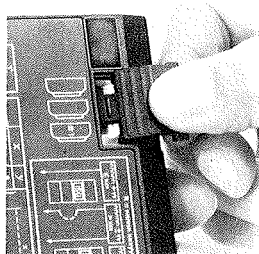
Hvis der går mere end ½ minut, risikerer De dog at slette den kontinuerlige hukommelse.

Batterierne udskiftes på følgende måde:

1. Sørg for at maskinen er slukket.
2. Hold maskinen som vist på figuren og tryk fremad på batteridækslet, indtil det glider løs.



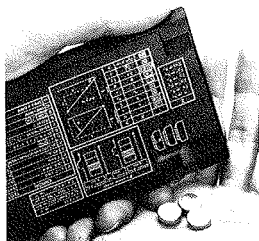
3. Tag fat i batteridækslets frie ende og løft dækslet af.



BEMÆRK

Pas på, at De ikke kommer til at trykke på nogen af HP-15C's taster under udførelsen af resten af batteriskiftet. Hvis De kommer til at aktivere en tast, risikerer De, at den kontinuerlige hukommelse bliver slettet.

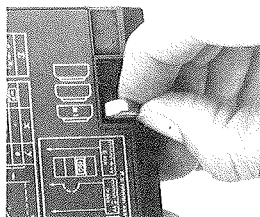
4. Vend regnemaskinen om og slå den forsigtigt mod håndfladen, så batterierne falder ud.



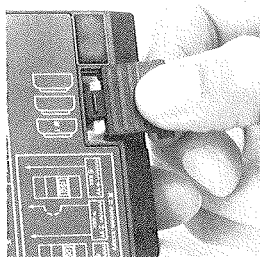
BEMÆRK

Pas på, at De ikke sætter batterierne omvendt på plads. Hvis det sker, risikerer De at slette den kontinuerlige hukommelse.

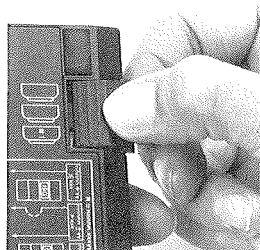
5. Sæt tre nye batterier på plads i batterirummet og sørg endelig for, at batterierne bliver placeret som vist på tegningen ud for batterirummet. +-polerne (de flade sider) skal vende op mod metalkontakten.



6. Skub batteridækslets lange tap ind i slidsen under kabinettets bagside.



7. Lad dækslet falde til og pres det ind mod kabinettet, indtil det sidder helt fast.
8. Tryk på **ON**, så der tændes for maskinen. Hvis den kontinuerlige hukommelse af en eller anden grund er blevet slettet, vises **Pr Error** i lyspanelet.



Selvtest

Hvis det virker, som om maskinen ikke fungerer rigtigt eller ikke vil tænde, kan De forsøge følgende:

Hvis maskinen ikke reagerer på indtastninger:

1. Tryk samtidigt på ☐Y⁺ og ☐ON og slip derefter begge taster. Dette vil ændre indholdet af X-registret, som derfor skal nulstilles bagefter.
2. Hvis maskinen stadig ikke reagerer på indtastninger, skal De først tage batterierne ud og sætte dem på plads igen. Sørg for, at batterierne sidder rigtigt i batterirummet: med pluspolerne rettet op mod metalkontakten i batterirummet. Hvis det ikke hjælper, bør De udskifte batterierne til nogle, som De ved er i orden.
3. Hvis maskinen stadig ikke reagerer på indtastninger, skal De lade batterierne sidde i batterirummet og kortslutte polerne. (Fold plastic-folien tilbage, så polerne – de flade metalstykker yderst i batterirummet – bliver frilagt.)

Kortslutningen skal kun vare et kort øjeblik. Efter kortslutningen vil indholdet af den kontinuerlige hukommelse være tabt, og De skal trykke på ☐ON for at starte maskinen.

4. Hvis heller ikke dette hjælper, skal maskinen til reparation hos Hewlett-Packard.

Hvis maskinen reagerer på indtastninger udføres følgende:

1. Sluk for HP-15C, tryk på ☐ON, hold den nedtrykket og tryk samtidig på ☐X.
2. Slip først ☐ON og dernæst ☐X. Herved startes en automatisk selvtest af alle HP-15C's elektroniske kredse.

Hvis alt er i orden, vil lommeregneren efter ca. 25 sekunders forløb (hvor **running** vises i lyspanelet) stoppe selvtesten og udlæse -8,8,8,8,8,8,8,8,8,8, samt alle lyspanelets statusindikatorer (undtagen ★) – og indikatorer som normalt ikke kan vises, vil blive vist

efter denne test. Hvis der udlæses **Error 9**, hvis lyspanelet er tomt, eller der kommer et andet resultat, skal maskinen sendes til reparation hos Hewlett-Packard.*

Bemærk: Også $\boxed{+}$ og $\boxed{\div}$ udfører interne checkrutiner, når de bruges på ovennævnte måde sammen med $\boxed{\text{ON}}$.** Disse tests bruges til at kontrollere kredsløbene under fremstillingsprocessen og ved service.

Hvis De har haft mistanke til maskinen, men trin 2 giver den rette udlæsning, er der stor sandsynlighed for, at De har fejlbetjent maskinen. Vi foreslår, at De genlæser det relevante afsnit i brugsanvisningen. Hvis De stadig får resultater, der ikke kan være rigtige, beder vi Dem sende maskinen til eftersyn hos Hewlett-Packard.

Udbedring af mangler

Såfremt der konstateres materiale- og/eller fabrikationsmangler, forpligter Hewlett-Packard sig til uden beregning at udbedre sådanne mangler på betingelse af, at der rettidigt er reklameret herover inden for en periode på tolv måneder fra leveringsdagen, og at lommeregneren returneres med forsendelsesomkostningerne betalt til et af Hewlett-Packard udpeget servicecenter sammen med dokumentation for købsdatoen.

* Hvis $\boxed{\text{ON}}/\boxed{\times}$ eller $\boxed{\text{ON}}/\boxed{+}$ giver **Error 9**, kan De fortsætte med at bruge maskinen efter at have slettet den kontinuerlige hukommelse.

** $\boxed{+}/\boxed{\text{ON}}$ starter en test, der svarer til den ovennævnte, men som blot fortsætter, indtil De stopper den ved at trykke på en vilkårlig tast (der kan gå op til 25 sekunder, før testen afbrydes efter tryk på en tast).

$\boxed{\text{ON}}/\boxed{\div}$ bruges til en test af lyspanel og tastatur. Når $\boxed{\text{ON}}$ slippes, tændes nogle enkelte af lyspanelets segmenter. Testen udføres ved, at man trykker på alle tasterne, en efter en, i en bestemt rækkefølge: man starter med tasten yderst til venstre i øverste række, fortsætter med den næste tast i samme række. Således fortsætter man, indtil alle taster er brugt. Hvis maskinen virker, og man har trykket på tasterne i den rigtige rækkefølge, vil der stå **15** i lyspanelet, når testen er forbi. ($\boxed{\text{ENTER}}$ skal bruges både under test af tredje række og under fjerde række.) Hvis maskinen ikke virker efter hensigten, eller hvis tasterne ikke er trykket i den rigtige rækkefølge (og det er langt mere sandsynligt), vil **Error 9** blive vist i lyspanelet.

Hvis De får udlæst **Error 9** efter denne test, bør De gennemgå den mindst én gang til: det sker meget let, at man glemmer en tast, trykker to gange på en tast eller på anden måde gør noget forkert.

Transportskader og forsendelsesomkostninger dækkes ikke. Foranstående forpligtelser gælder ikke, hvis lommeregneren er beskadiget ved uheld, misbrug eller på grund af service eller ændringer foretaget af andre end et autoriseret Hewlett-Packard servicecenter. Eventuelle mangler på forbrugsartikler dækkes ikke af foranstående forpligtelser. *Hewlett-Packard er ikke ansvarlig for følgevirkninger.*

Specifikationer

Hewlett-Packards produkter sælges på basis af de specifikationer, der er gældende på salgstidspunktet. Hewlett-Packard har ingen forpligtelse til at ændre eller modernisere produkter, der er solgt og betalt.

International serviceorganisation

Ikke alle Hewlett-Packard servicecentre kan yde service på alle typer lommeregnerne fra Hewlett-Packard. De kan dog være sikker på at kunne få repareret Deres lommeregner i det land, hvor den er købt.

Hvis De opholder Dem uden for indkøbslandet, kan De kontakte det lokale Hewlett-Packard kontor, hvor det vil kunne oplyses, om lommeregneren kan repareres på stedet.

Alle de nedennævnte Hewlett-Packard servicecentre i Europa kan udføre service på alle lommeregnerne fra Hewlett-Packard.

Også for service i et andet land end indkøbslandet gælder de ovennævnte forpligtelser. Det bemærkes specielt, at De skal kunne dokumentere købsdatoen for at kunne opnå vederlagsfri udbedring af mangler.

Servicecentre i Europa

BELGIEN

HEWLETT-PACKARD BELGIUM SA/NV

Boulevard de la Woluwe 100

Woluwelaan

B-1200 BRUXELLES

Telefon (2) 762 32 00

DANMARK

HEWLETT-PACKARD A/S

Datavej 52

DK-3460 BIRKERØD

Telefon (02) 81 66 40

FINLAND

HEWLETT-PACKARD OY

Revontulentie 7

02100 ESPOO 10 (Helsinki)

Telefon (90) 455 02 11

FRANKRIG

HEWLETT-PACKARD FRANCE

S.A.V. Calculateurs de Poche

Division Informatique Personnelle

F 91947 LES ULIS CEDEX Telefon (6) 907 78 25

HOLLAND

HEWLETT-PACKARD NEDERLAND B.V

Van Heuven Goedhartlaan 121

NL 1181 KK AMSTELVEEN (Amsterdam)

P.O. Box 667

Telefon (020) 472021

ITALIEN

HEWLETT-PACKARD ITALIANA S.P.A.

Casella postale 3645 (Milano)

Via G. de Vittorio, 9

I 20063 CERNUSCO SUL NAVIGLIO (Milano)

Telefon (2) 90 36 91

NORGE

HEWLETT-PACKARD NORGE A/S

P.O. Box 34

Østerndalen 18

N 1345 ØSTERÅS (Oslo)

Telefon (2) 17 11 80

SCHWEIZ

HEWLETT-PACKARD (SCHWEIZ) AG

Kleinrechner-Service

Allmend 2

CH 8967 WIDEN

Telefon (057) 50111

SPANIEN**HEWLETT-PACKARD ESPANOLA S.A.**

Calle Jerez 3

E MADRID 16

Telefon (1) 458 2600

SVERIGE**HEWLETT-PACKARD SVERIGE AB**

Enighetsvägen 3

Box 205 02

S 161 BROMMA 20 (Stockholm)

Telefon (8) 730 05 50

STORBRITANNIEN**HEWLETT-PACKARD Ltd**

King Street Lane

Winnersh, Wokingham

GB BERKSHIRE RG11 5 AR

Telefon (734) 784774

VESTTYSKLAND**HEWLETT-PACKARD GmbH**

Kleinrechner-Service

Vertriebszentrale

Berner Strasse 117

Postfach 560 140

D 6000 FRANKFURT 56

Telefon (611) 50041

ØSTLANDENE

Se adressen under ØSTRIG

ØSTRIG**HEWLETT-PACKARD GmbH**

Kleinrechner-Service

Wagramerstr.-Lieblgasse

A 1220 WIEN

Telefon (222) 23 65 11

Efter udbedringsperiodens udløb

Efter udbedringsperioden på ét år reparerer lommeregneren til lavest mulige pris. Der er tre måneders forpligtelsesperiode på alle reparationer, der udføres efter udbedringsperiodens udløb af et autoriseret Hewlett-Packard servicecenter.

Overførsel af ovennævnte forpligtelse

Hvis De sælger Deres lommeregner eller forærer den væk, kan ovennævnte forpligtelser overdrages og gælder stadig for den ny ejer, indtil den oprindelige ét-årige forpligtelsesperiode er udløbet. Det er ikke nødvendigt at give Hewlett-Packard meddelelse om overdragelsen. Dokumentation for købet (evt. kvittering) bør dog videregives til den ny ejer.

Forsendelsesvejledning

Lommeregneren returneres sammen med det udfyldte servicekort i salgskartonen (eller anden emballage, der yder tilsvarende beskyttelse) for at undgå skader under transporten. Transportskader dækkes ikke af Hewlett-Packard, som derfor anbefaler, at alle forsendelser forsikres særskilt. Hvis lommeregneren skal repareres vederlagsfrit i henhold til ovennævnte forpligtelse, skal De vedlægge dokumentation for købsdatoen.

Lommeregneren skal returneres til nærmeste af Hewlett-Packard udpegede servicecenter. De bedes kontakte Deres HP-forhandler for nærmere oplysninger.

Yderligere oplysninger

Oplysninger om lommeregnerens kredsløb og design betragtes som Hewlett-Packards ejendom, ligesom servicehåndbøger ikke stilles til Deres rådighed. Hvis der skulle opstå andre problemer, bedes De kontakte Deres HP-forhandler eller det nærmeste Hewlett-Packard kontor.

Temperaturspecifikationer

- Brugstemperatur: 0° til 55°C
- Lagertemperatur: – 40° til 65°C

Oversigt over funktionstaster

Dataregisterfunktioner	265
Forvalgstaster	266
Hyperbolske funktioner	266
Indeksregister, kontrol af	266
Komplekse funktioner	266
Logaritme- og eksponentialfunktioner	267
Matematiske funktioner	267
Matrixfunktioner	267
Omregningsfunktioner	269
Procentfunktioner	269
Sandsynlighedsregning	269
Stakmanipulation	270
Statistik	270
Talindtastning	271
Trigonometriske funktioner	271
Udlæsningsformatet, kontrol af	271
Ændring af det udlæste tal, funktioner til	271

Bemærk:

I alle tilfælde er x tallet i X-registret, y tallet i Y-registret og så videre.

ON

ON Tænder og slukker for lyspanelet (side 18). Bruges desuden til nulstilling af den kontinuerlige hukommelse (side 62), til skift af decimaltegn (side 60) og i forbindelse med de forskellige selvtests (side 259-260).

Dataregisterfunktioner

STO Lagring. Lagrer en kopi af et tal i det specificerede dataregister {0 til 9, .0 til .9, **(i)**, **1**} (side 42). Bruges desuden til registeraritmetik: nyt registerindhold = oprindelige registerindhold { **+**, **-**, **×**, **÷** } lyspanelets indhold (side 43).

RCL Fremkald.

Fremkalder en kopi af tallet i det speci-

cerede dataregister {0 til 9, .0 til .9, **(i)**, **1**} (side 42). Bruges desuden til registeraritmetik: nyt lyspanelindhold = oprindelige lyspanelindhold { **+**, **-**, **×**, **÷** } registerindhold (side 43).

CLEAR REG Nulstiller alle dataregistre (side 43).

LSTx Fremkalder det sidst brugte argument (side 35).

Forvalgstaster

[f] Bruges før en funktionstast til at vælge den funktion, som er trykt med gult oven over tasten (side 18).

[g] Bruges før en funktionstast til at vælge den funktion, som er trykt med blå på tastens skrå forside (side 18).

Se desuden afsnittene om taster til kontrol af lyspanelet (side 271), dataregistertaster (side 265) og oversigten over programmeringstaster (side 272).

CLEAR [PREFIX] Annullerer enhver brug af en forvalgstast eller delvist indtastet instruktion (som f.eks. [f] [SCI]) (side 19). Udlæser desuden mantissens 10 mest betydende cifre (side 59).

Hyperbolske funktioner

[HYP] [SIN], [HYP] [COS], [HYP] [TAN]

Beregner henholdsvis hyperbolsk sinus, cosinus og tangens (side 28).

[HYP⁻¹] [SIN], [HYP⁻¹] [COS], [HYP⁻¹] [TAN]

Beregner henholdsvis invers hyperbolsk sinus, cosinus og tangens (side 28).

Indeksregister, kontrol af

[I] Indeksregister (R₁). Dataregister som bruges til: indirekte programudførelse og forgrening med [GTO] og [GSB], kørsel af løkker med [ISG] og [DSE], indirekte flagkontrol og indirekte formattering af lyspanelet (side 106). Bruges desuden ved indtastning af komplekse tal og til aktivering af kompleksstatus (side 119).

(i) Indirekte operationer. Bruges til at adressere *et andet* dataregister *ved hjælp af* R₁ i forbindelse med lagring, fremkald, aritmetik og løkke-

kontrol (side 106). Bruges desuden med [DIM] til at ændre splitningen (side 212).

Komplekse funktioner

[Re $\geq$ Im] Ombytning af reel- og imaginærdel. Aktiverer kompleksstatus (ved at oprette en imaginær stak) og ombytter indholdene af den reelle og den komplekse staks X-registre (side 122).

[I] Bruges ved indtastning af komplekse tal. Aktiverer kompleksstatus (ved at oprette en imaginær stak) (side 119). Bruges desuden sammen med [DIM] til indirekte dimensionering af matricer (side 171). (Se afsnittet om kontrol af indeksregistret side 266 for information om brugen i forbindelse med indeksregistret).

(i) Udlæser indholdet af det imaginære X-register så længe ta-

sten holdes nedtrykket (**side 122**).

[SF] 8 Sætter flag 8 og aktiverer kompleksstatus (**side 119**).

[CF] 8 Sletter flag 8 og ophæver kompleksstatus (**side 119**).

Logaritme- og eksponentialfunktioner

[LN] Beregner den naturlige logaritme af x (**side 27**).

[e^x] Eksponentialfunktionen, som er antilogaritme til den naturlige logaritme. Opløfter e til x 'te potens, hvor x er tallet i X-registret (**side 28**).

[LOG] Beregner 10-talslogaritmen af x (**side 28**).

[10^x] Almindelig antilogaritme. Opløfter 10 til x 'te potens (**side 28**).

[y^x] Opløfter y til x 'te potens, hvor x og y er tallene i X og Y. Stak-

ken falder efter denne funktion (**side 29**).

Matematiske funktioner

[+], [-], [×], [÷] Aritmetiske operatoren; får stakken til at falde (**side 29**).

[$\sqrt{x}$] Beregner kvadratroden af x (**side 25**).

[x^2] Kvadrerer x (**side 25**).

[$x!$] Beregner fakultet ($n!$) af x eller gammafunktionen (Γ) af $(1+x)$ (**side 25**).

[$1/x$] Beregner reciprok-værdien af x (**side 25**). (Se afsnittet om matrixfunktioner, **side 267**, for information om brugen i forbindelse med matricer).

[π] Lagrer værdien af π i X (**side 24**).

[SOLVE] Finder et reelt nulpunkt for funktionen $f(x)$, hvor

$f(x)$ er defineret ved hjælp af et underprogram (**side 178**).

[$\int$] Numerisk integration. Bestemmer en tilnærmelse til det bestemte integral af $f(x)$, hvor $f(x)$ er defineret ved hjælp af et underprogram (**side 191**).

Matrixfunktioner

Bemærk: Ved matricen i X forstås den matrix hvis deskriptor er lagret i X.

[DIM] Dimensionerer en navngiven matrix { **[A]** til **[E]**, **[I]** } (**side 139**).

[RESULT] Angiver hvilken matrix der skal bruges til at lagre resultatet af visse matrixoperationer (**side 146**).

[USER] I USER-status bliver række/søjletallene i R_0 og R_1 automatisk justeret til næste matrixelement hver gang **[STO]** eller **[RCL]** { **[A]** til **[E]**,

[(i)] } bliver brugt
(side 142).

[STO] eller **[RCL]** { **[A]**
til **[E]**, **[(i)]** } Lagrer
eller fremkalder ma-
trixelementet hvis
række/søjlenummer
er lagret i R_0 og R_1
(siderne 142 og
144).

[STO] **[g]** eller **[RCL]**
[g] { **[A]** til **[E]**, **[(i)]** }
Lagrer eller fremkal-
der matrixelementet
hvis række/søjlenum-
mer er lagret i Y og X
(side 144).

[STO] eller **[RCL]**
[MATRIX] { **[A]** til
[E] } Lagrer eller
fremkalder den speci-
ficerede matrix (si-
derne 140 og 145).

[STO] og **[RCL]**
[RESULT] Lagrer eller
fremkalder matricen,
der bruges som resul-
tatmatrix (side 146).

[RCL] **[DIM]** { **[A]** til
[E], **[I]** } Udlæser di-
mensionen for den
specifiserede matrix;
antal rækker i Y og
antal søjler i X (side
140).

[1/x] Inverterer ma-
tricen i X og lagrer
resultatet i den valgte
resultatmatrix, hvis
deskriptor herefter
udlæses (side 148).

[+], **[-]**, **[×]** Udfører
addition, subtraktion
eller multiplikation
mellem korresponde-
rende elementer i to
matricer eller mellem
en skalar og en ma-
trix. Resultatet lagres
i resultatmatricen
(side 150-153).

[÷] Når operatoren
virker på to matricer,
bliver den inverse af
matricen i X matrix-
multipliseret med
matricen i Y . Når
operatoren virker på
en enkelt matrix i Y ,
vil hvert af elemen-
terne blive divideret
med skalamet i X . Når
operatoren virker på
en enkelt matrix i X ,
bliver hvert enkelt
element i den inverse
af X multipliseret
med skalamet i Y
(side 150-153).

[CHS] Skifter fortegn
på alle elementer i

matricen i X (side
148).

[MATRIX] {0 til 9}
Matrixoperationer.
[MATRIX] 0 Dimensio-
nerer alle matricer til
 0×0 (side 141).

[MATRIX] 1 Lagrer 1 i
såvel R_0 som R_1 (side
141).

[MATRIX] 2 Komplex
transformation: Z^P til
 $\bar{Z}$ (side 162).

[MATRIX] 3 Invers
kompleks transfor-
mation: $\bar{Z}$ til Z^P (side
162).

[MATRIX] 4 Transpo-
nering: X til $Y^T X$
(side 148).

[MATRIX] 5 Transpo-
nering med multipli-
kation: Y og X til $Y^T X$
(side 152).

[MATRIX] 6 Beregner
og lagrer restmatricen
i resultatmatricen
(side 157).

[MATRIX] 7 Beregner
rækkenormen for
matricen i X (side
148).

MATRIX 8 Beregner den euklidiske norm for matricen i X **(side 148)**.

MATRIX 9 Beregner determinanten for matricen i X, som samtidig *LU*-faktoriiseres **(side 148)**.

Cy,x Transformerer matrix på partitioneret form (Z^P) til kompleks form (Z^C) **(side 160)**.

Py,x Transformerer matrix på kompleks form (Z^C) til partitioneret form (Z^P) **(side 160)**.

x=0, **TEST** 0, **TEST** 5, **TEST** 6 Betingede testoperationer for matricer i X eller i X og Y. **x=0** og **TEST** 0 ($x \neq 0$) tester om størrelsen i X er nul.

Matrixdeskriptorer er pr. definition forskellige fra nul. **TEST** 5 ($x = y$) og **TEST** 6 ($x \neq y$) tester om deskriptorerne i X og Y er identiske. Hvis te-

sten er falsk bliver næste programlinje sprunget over **(side 171)**.

Omregningsfunktioner

R Omregner de polære koordinater r og θ i henholdsvis X og Y til rektangulære koordinater x og y **(side 30)**. Se desuden afsnittet om kompleksstatus **side 132**.

P Omregner de rektangulære koordinater x og y i henholdsvis X og Y til polære koordinater r og θ **(side 30)**. Se desuden afsnittet om kompleksstatus **side 132**.

H.MS Omregner decimaltimer (eller grader) til timer, minutter og sekunder (eller grader, minutter og sekunder) **(side 27)**.

H Omregner timer, minutter og sekunder (eller grader, minutter og sekunder) til decimaltimer

(eller grader) **(side 27)**.

RAD Omregner grader til radian **(side 27)**.

DEG Omregner radian til grader **(side 27)**.

Procentfunktioner

% Procent. Beregner $x\%$ af tallet i Y **(side 29)**. I modsætning til de fleste andre funktioner af to variable får procent ikke stakken til at falde.

Δ% Procentisk forskel. Beregner den procentiske forskel mellem tallet i Y og tallet i X **(side 29)**. Stakken falder ikke.

Sandsynlighedsregning

Cy,x Kombinatorer. Beregner antallet af kombinationer af y forskellige genstande taget x ad gangen **(side 47)**. Stakken falder. (Se desuden afsnittet om Matrixfunktioner **side 267**).

[Py,x] Permutationer. Beregner antallet af permutationer af y forskellige genstande taget x ad gangen **(side 47)**. Stakken falder. (Se desuden afsnittet om Matrix-funktioner **side 267**).

Stakmanipulation

[x $\approx$ y] Ombytter indholdene af X og Y **(side 34)**.

[x $\approx$] Ombytning med x . Ombytter tallet i X med tallet i et andet, specificeret register. Kan bruges sammen med **[il]**, **[I]** og adresser med **[·]** **(side 42)**.

[Re $\approx$ Im] Ombytning af reel- og imaginærdel. Ombytter indholdet af det reelle X-register og det imaginære og aktiverer kompleksstatus **(side 122)**.

[R ∇] Ruller stakken ned – så y kommer til X osv. **(side 34)**.

[R $\uparrow$] Ruller stakken op – så x kommer til Y osv. **(side 34)**.

[CLx] Sletter (nulstiller) lyspanelet (X-registret) **(side 21)**.

[$\leftarrow$] I normalstatus: sletter det sidst indtastede ciffer. Sletter dog hele det udlæste tal, hvis talindtastning er afsluttet **(side 21)**.

Statistik

[$\Sigma+$] Summerer tallene fra X og Y i dataregistrene R₂ til R₇ **(side 49)**.

[$\Sigma-$] Annullerer summation i dataregistrene R₂ til R₇ af tallene i X og Y **(side 51)**.

[$\bar{x}$] Beregner middelværdierne af de x - og y -værdier, der er summeret med **[g]** **[$\Sigma+$]** **(side 53)**.

[s] Beregner stikprøvestandardafvigelse for de x - og y -værdier, der er summeret med **[g]** **[$\Sigma+$]** **(side 53)**.

[$\hat{y},r$] Lineært estimat og korrelationskoefficient. Beregner et

y -estimat ($\hat{y}$) for en given værdi af x ved hjælp af mindste kvadraters metode og lagrer resultatet i X. Samtidig beregnes korrelationskoefficienten, r , for de summerede data. r lagres i Y **(side 55)**.

[L.R.] Lineær regression. Bruger mindste kvadraters metode til at beregne skæringspunkt på y -akse og hældningskoefficient for den rette linie, der bedst tilnærmer data-sættet som er summeret med **[$\Sigma+$]**. Værdien for skæringspunktet lagres i X medens værdien for hældningskoefficienten lagres i Y **(side 54)**.

[RAN#] Pseudotilfældige tal. Udlæser et pseudotilfældigt tal på basis af et starttal, som lagres med **[STO]** **[RAN#]** **(side 48)**.

CLEAR **[Σ]** Sletter (nulstiller) indholdet af statistikregistrene (R₂ til R₇) **(side 49)**.

Talindtastning

[ENTER] Lagrer en kopi af X-registrets indhold i Y. Bruges til at adskille separate talindtastninger (**siderne 22, 37**).

[CHS] Skifter fortegn på den udlæste mantisse eller eksponent (**siderne 19, 122**).

[EEX] Indtastning af eksponent. De følgende cifre betragtes som eksponentcifre (**side 19**).

[0] til **[9]** Taltaster (**side 22**).

[.] Decimaltegn (**side 22**).

Trigonometriske funktioner

[DEG] Vælger grader som vinkelmål i forbindelse med de trigonometriske funktioner. Har ingen særskilt markering i lyspanelets symbolrække (**side 26**). Kan ikke bruges sammen med komplekse argumenter.

[RAD] Vælger radian som vinkelmål i forbindelse med de trigonometriske funktioner. Markeres af RAD i lyspanelets symbolrække (**side 26**).

[GRD] Vælger nygrader som vinkelmål i forbindelse med de trigonometriske funktioner. Markeres af GRAD i lyspanelets symbolrække (**side 26**). Kan ikke bruges sammen med komplekse argumenter.

[SIN], **[COS]**, **[TAN]**
Beregner henholdsvis sinus, cosinus og tangens af x (**side 26**).

[SIN⁻¹], **[COS⁻¹]**,
[TAN⁻¹] Beregner henholdsvis Arcus sinus, Arcus cosinus og Arcus tangens af x (**side 26**).

Udlæsningsformatet, kontrol af

[FIX] Vælger fast-komma som udlæsningsformat (**side 57**).

[SCI] Vælger flydende tals notation som udlæsningsformat (**side 58**).

[ENG] Vælger metrisk notation som udlæsningsformat (**side 58**).

Mantisse. **[f]** **CLEAR**
[PREFIX] udlæser alle 10 betydende cifre i det tal, der er lagret i X, indtil **CLEAR**
[PREFIX] slippes (**side 59**). Sletter desuden enhver tastsekvens, der ikke er fuldført (**side 19**).

Ændring af det udlæste tal, funktioner til
[ABS] Erstatter det udlæste tal med dets numeriske værdi (**side 24**).

[FRAC] Erstatter det udlæste tal med decimaldelen af tallet (**side 24**).

[INT] Erstatter det udlæste tal med heltalsdelen af tallet (**side 24**).

RND Afrunder tallet
i X-registret, så det
præcis svarer til det
udlæste tal (**side 24**).

Oversigt over programmeringstaster

[P/R] Programmeringsstatus/Normalstatus (*Program/Run*). HP-15C er i programmeringsstatus, hvor indtastninger ikke udføres men registreres som programinstruktioner, når PRGM i lyspanelets symbolrække er tændt. Ellers er maskinen i normalstatus, hvor indtastninger udføres (**side 64**).

CLEAR [PRGM] I programmeringsstatus: Sletter hele programhukommelsen og normalstiller splitningen. I normalstatus: Sætter programpil til linie 000 (**side 65**).

[MEM] Udlæser den aktuelle splitning (antallet af dataregistre, fællesregistre og programregistre) (**side 212**).

[◀] Baktast. I programmeringsstatus sletter denne tast den udlæste programlinje fra hukommelsen.

Alle efterstående linier i hukommelsen rykker én linie op (**side 82**).

[LBL] Etiket. Bruges sammen med en adresse til at markere starten på et særskilt programafsnit (**side 65**).

[A] [B] [C] [D] [E] 0 1
2 3 4 5 6 7 8 9 .0 .1 .2
.3 .4 .5 .6 .7 .8 .9 Etiketadresser (navne). Når de følger efter **[LBL]** definerer de starten af et særskilt programafsnit. Bruges desuden (uden **[LBL]**) til at starte udførelsen af et specifikt programafsnit eller underprogram (**side 67**).

[USER] Ind- og udkobler USER-status, hvor der byttes om på placeringen af de primære (**[A]** til **[E]**) og de gule sekundære funktioner på de fem første taster i øverste række (**side 67**).

USER-status har desuden indvirkning på matrixbrugen af

[STO] eller **[RCL]** { **[A]** til **[E]**, **(i)** }. I

USER-status opdateres indholdet af R_0 (rækkenummer) og R_1 (søjlenummer) automatisk, når **[STO]** eller **[RCL]** anvendes (**side 142**).

[GTO] Ubetinget hop. Bruges med en etiketadresse (se ovenfor) eller **[I]** til at føre programpil til det angivne programafsnit. Hvis funktionen bruges under programkontrol, fortsætter kørslen fra etiketten. Hvis funktionen bruges fra tastaturet, placeres programpil ud for etiketten (**side 89**). Hvis der er lagret et negativt tal i R_1 , vil **[GTO] [I]** resultere i hop til et linienummer (**side 108**).

[GTO] [CHS] nnn

Ubetinget hop til linienummer. Placerer

programpilens ud for linien med det angivne nummer. Kan ikke programmeres **(side 81)**.

[GSB] Hop til underprogram. Bruges med en etiketadresse (se ovenfor) eller **[I]** til at starte udførelsen af det angivne programafsnit (underprogram). Kan bruges under programkontrol og fra tastaturet. **[RTN]** fører program-pilen hen til linien umiddelbart efter den sidst udførte **[GSB]** **(side 100)**.

[BST] Bakfunktionen. Flytter program-pilen baglæns i program, én linie ad gangen, så længe tasten holdes nedtrykket. Herved udlæses linienummer og tastkoder for de enkelte linier **(side 82)**.

[SST] Skridtfunktionen. Flytter program-pilen forlæns i et program, én linie ad gangen, i programmeringsstatus så længe

tasten holdes nedtrykket. I normalstatus udlæses linienummer og tastkoder medens tasten holdes nedtrykket. Når den slippes, udføres den lagrede instruktion **(side 81)**.

[PSE] Pause. Stopper programudførelsen i ca. 1 sekund medens indholdet af X-registret bliver udlæst. Herefter fortsættes programudførelsen med næste programlinie **(side 66)**.

[R/S] Kør/Stop. Hvis programmet ikke kører, vil brug af tasten starte programkørslen fra programpilens aktuelle position. Hvis programmet kører, vil det stoppe, når den aktuelle instruktion er udført **(side 66)**.

[RTN] Retur. Virker normalt som **[R/S]**, dog med hop til linie 000 **(side 66)**. Hvis **[RTN]** mødes under udførelsen af et underprogram, vil

program-pilen blive ført til linien umiddelbart efter den sidst brugte **[GSB]** **(side 100)**.

[SF] Sæt flag (= sand). Sætter det specificerede flag (0 til 9). Flagene 0 til 7 er brugerflag, flag 8 markerer kompleksstatus og flag 9 bruges til at markere overløb **(side 91)**.

[CF] Slet flag (= falsk). Sletter det specificerede flag (0 til 9) **(side 91)**.

[F?] Er flaget sat? Tester om det specificerede flag er sat. Hvis flaget er sat, fortsætter programudførelsen direkte; hvis flaget er slettet, bliver næste programlinie sprunget over, før kørslen fortsætter **(side 91)**.

[$x \leq y$], [$x = 0$], [TEST] {0 til 9} Sammenligningsoperationer (betingede hop). Hver af disse operationer sammenligner tallet i X med enten 0 eller

tallet i Y. Hvis testen er opfyldt (sand), fortsætter kørslen direkte. Hvis testen ikke er opfyldt (falsk), bliver næste programlinie sprunget over, før kørslen fortsætter **(side 90)**.

$x=0$ og **TEST** 0, 5 og 6 kan desuden bruges til at sammenligne komplekse tal og matrixdeskriptorer **(siderne 130, 171)**.

TEST 0 $x \neq 0$

TEST 1 $x > 0$

TEST 2 $x < 0$

TEST 3 $x \geq 0$

TEST 4 $x \leq 0$

TEST 5 $x = y$

TEST 6 $x \neq y$

TEST 7 $x > y$

TEST 8 $x < y$

TEST 9 $x \geq y$

DSE Dekrementering og skip hvis lig med eller mindre end. Trækker et inkrement fra (*dekrementerer*) løkketælleren og skipper den efterføl-

gende linie, hvis tælleren herefter er lig med eller mindre end en given testværdi **(side 108)**.

ISG Inkrementering og skip hvis lig med eller større end. Lægger et inkrement til (*inkrementerer*) løkketælleren og skipper den efterfølgende linie, hvis tælleren herefter er lig med eller større end en given testværdi **(side 108)**.

Stikordsregister

Sidenumre i **halvfed** angiver primære referencer, medens sidenumre i normal skrift angiver sekundære referencer.

A

Afrunding

fejl ved brug af **SOLVE**, **220**, 234

fejl 52, **59**

lyspanel, **58**

RND, **24**

Aktivering af stakløft, **36**

Andengradsligning, løsning, **179**

Antilogaritme, 10-tals og naturlig, **28**

Aritmetiske operationer, **28**, 37

Assistance, teknisk, **270**

Asymptote, vandret, **227**

Automatisk justering af række/søjlenumre, **141**

B

Bakfunktion **BST**, **82**

Bakteriepopulation, eksempel, **41**

Batterilevetid, **255**

Batteriudskiftning, 256, **257-258**

Bessels funktion, **194**, 196

Betingelsestest, se Betinget operation

Betinget forgrening, indirekte, **108-110**, 111, 115

Betinget operation

definitioner, **90**, 97, 189

kompleksstatus, **130**

matrixdeskriptorer, **171**

Blokering af stakløft, **36**

Boldkast, eksempler, **182-183**, 221-223

Brugerflag, **91**

C

C indikator, 99, 119

[CHS], 19

Cifferopdeling i lyspanel, **60**

[COS], [COS⁻¹], 26

D

Datalagring, **42**

Dataregisterblok, **210-211**

Dataregistre

aritmetik, **43**

definitioner, **42**

sletning, **43**

splitning, **42, 212-214**

statistik, **42, 49**

Decimaldel **[FRAC], 24**

Decimaltegn

komma, **22**

punkt, **22**

valg af decimaltegn, **60**

[DEG], 26

Determinant, **148**

[DIM], 75-76, 212-214

Direkte registeraritmetik, **43**

[DSE], 108-110, 111, 115

Dåsevolumen og -arealer, eksempel, **69-72**

E

[EEX], 19

Ekspionenter, **19-20**

Ekspontionalfunktionen, **28**

Elektrisk kredsløb, eksempel, **167-170**

[ENG], 57

[ENTER]

generelt, **12, 33-34, 37**

indflydelse på stakbevægelser, **36, 41**

indflydelse på talindtastning, **22**, 28
 Etiketter, **65**, **76**, 89, 96
 Euklidisk norm, 148, **174**

F

f, **18**

Fakultet **x!**, **25**

Fasenotation, **131**

Fastkomma, **57**

Fejl

f, **200-201**

fejltilstande, **202-205**

SOLVE, **184**, 190

stop, **77**

udlæsning, **60**

FIX, **57**

Flag 8, **99**

Flag 9, **99**

Flagtest, **91**, 98

Flydende tal, **58**

Forgrening

betinget, **90**, 97, 174, 189

indirekte, **107-108**, 111-113, 114

ubetinget, **89**

Forkortet indtastning, **77**

Formater i eksempler, 2, **18**

Forskydningsspænding, **225-226**

Fortegnsskift

kompleksstatus, **122-123**

mantisseeksponent, **19**

matricer, **174**

Forvalgstaster, **19**

Fremkald

aritmetik, **44**

matricer, **142**, 147, 173

RCL, **42**, 44

statistikdata, **50**

Frie registre, 210, **212**, 214

Frit fald, eksempel, **14**

Funktioner

alternative, **18**

én variabel, 22, **25**

ikke-programmérbare, **79**

primære og alternative (sekundære), **18**

sekundære, **18**

to variable, 22, **28-29**

Fællesblok, **210**

G

$\boxed{g}$, **18**

Gammafunktionen $\boxed{x!}$, **25**

$\boxed{GRD}$, **26**

$\boxed{GSB}$, **100**

$\boxed{GTO}$, **89**, 96, 97

$\boxed{GTO}$ $\boxed{CHS}$, **81**

Gæld, eksempel, **94**

II

Heltalsdel $\boxed{INT}$, **24**

HOP hvis JA, **91**, 189

HOP hvis SAND, **91**, 189

Horner's metode, **78**, 179

Hukommelse

begrænsninger, 74, 76, **214**

fordeling, 74, **210-211**

fri lagerplads, 73-76, 210, **212**

krav ved brug af avancerede funktioner, **215-216**

krav ved programmering, **214-215**

registre, **210-212**

splitning, 73-74, **212-214**

stak (se Rullestak)

startsplitning, 73-74

statusudlæsning, **212**

Hyperbolske funktioner, **28**

Hædningskoefficient, **54**

Høstudbytte, eksempel, **154-157**



Ikke-programmérbare funktioner, **79**

Imaginærstak

oprettelse, **118-121**, 130

sletning, **122**

stakløft, **122**

udlæsning, **122**

Indeksregister

aritmetik, **107**, 111

flagkontrol, **108**, 114

lagring og fremkald, **106**, 110, 114

løkkekontrol, 106, **108-110**

ombytning med X-register, **107**, 111

udlæsningsformat, kontrol af, **108**, 113, 114, 115

Indikator for lav batterispænding, 61, **256**

Indikatorer

kompleks, **119**

liste, **59**

PRGM, 32, **66**

trigonometri, **25-26**

Indirekte adressering, **105-107**, 114

Indtastning

eksponent, **19-20**

funktioner af én variabel, **22**

funktioner af to variable, **22**, 28

kæderegning, **22**

Indtastning af statistikdata, **49**

Initialisering, **86**

Instruktioner, **73**

Integrering, se Numerisk integration

Interpolation med **$\overline{y}, r$** , **56**

ISG, **108-110**, 115

Iterationer med **ISG** og **DSE**, **110**

K

Koefficientmatrix, **154**

Kombinationer $\boxed{C_{y,x}}$, **47**

Kompleks aritmetik, eksempel, **130**

Kompleks matrix

invertering, 160, 162, **163**

lagring af elementer, **159**

multiplikation, 160, 162, **164**

transformation, **160, 162**

Komplekse tal

indtastning, **119, 125, 126-127**

konvertering af polære og rektangulære formater, **131-133**

lagring og fremkald, **128**

sletning, **123-125**

Kompleksstatus

aktivering af 99, **118-119, 130-131**

definitioner, **118-119**

matematiske funktioner, **129**

ophævelse af, **119**

stakløft, **122**

Konjugering, **123**

Konstant

aritmetiske udtryk, 35, **39-42**

beregninger, **39-42**

Konstanter i stakken, 39, **41**

Konstantmatrix, **154**

Kontinuerlig hukommelse

holdbarhed, **61**

nulstilling (sletning), **62**

omfang, 43, 48, 57, 60, **61**

Kontroltal i løkker, **108, 115**

Konventioner i brugervejledningen, **18**

Korrektion af statistikdata $\boxed{\Sigma-}$, **51-52**

Korrektion af summerede data, **51-52**

Korrelationskoefficient $\boxed{\hat{y},r}$, **55**

Kvadratrod $\boxed{\sqrt{x}}$, **25**

Kvadrering $\boxed{x^2}$, **25**

Kæderegning, **22-23**, 38

Kør/Stop R/S, **66**, 90

L

Lagring og fremkald STO og RCL

direkte (med I), 105, **106**

generelt, **42**, 43, 44

indirekte, **105-106**, 110

komplekse tal, **128**

matricer, **142**, 145, 147

matricelementer, **141-142**, 145, 147

Lineær regression, **54**

Lineære ligninger, **136**, 154

Lineært estimat $\hat{y}_i$, **55**

Logaritmefunktion, 10-tals og naturlig, **27-28**

LU-faktorisering, **146**, 153, 154, 158

Łukasiewicz, Jan, **32**

Lyspanel (se også X-register)

blinkende, **99**

fejlmeddelelser, **60**

kompleksstatus, **119**

mantis, **59**

sletning, **21**

Løkker, **89**, 97

M

Mantis, udlæsning af alle cifre, **59**

Matricer

deskriptorer, 137, **145-146**, 158

deskriptorer i R_1 , **171**

dimensionering, **138-141**, 171

hukommelse, **138-139**, 169

komplekse, **158-161**

komplekse ligninger, **166**

kopiering, **147**

navn (se Matrixdeskriptorer)

partitionering, **159**, 162

udlæsning af dimension, **140**, 145

Matricelementer

- individuel behandling, **143-145**
- lagring og fremkald, **141-142**, 145, 147, 173
- udlæsning, **142**

Matrixfunktioner

- aritmetik, **151**
- enkeltmatrix-, **147-149**
- euklidisk norm, 148, **174**
- invertering, **148**, 152
- multiplikation, **152**
- programmering, **173-174**
- reciprok, **148**
- registerbrug, **170-171**
- restmatrix, **157**
- resumé, **174-176**
- R_I , **171**
- rækkenorm, 148, **174**
- sammenligningsoperationer, **171**
- transponering, **148**, 149, 152

Mellemresultater, 22, 38**MEM**, **212****Metalkasse, eksempel, 186-188****Metrisk notation, 58****Middelværdi $\bar{x}$, 52-53****Minima med SOLVE, 227****Multiple rødder, 231**

N**Negative tal**

- komplekse tal, **122-123**
- reelle tal, **19**

Neutrale operationer, 208**Normaliserede statistiske data, 50****null, 142, 147****Nulstilling, se Sletning****Numerisk integration β**

- algoritme, 193, **237-238**, 245, 247, 251-252

beregningstid, 193, **197**, 241, 242, **250-252**
 foreløbig tilnærmelse, **252-253**
 funktionsunderprogram, **191-192**
 hukommelsen, brug af, **200-201**
 nøjagtighed, **197-200**, 237, **238-242**
 problemer med fluktuerende funktioner, **245-250**
 programmering, **200-201**
 rekursiv brug, **200**
 udlæsningsformatets betydning, **242-245**
 usikkerhed, **199-200**, 237-241, **242-245**
 Numerisk værdi ABS, **24**

O

Ombytning af imaginær og reel stak, **122**
 Omregninger
 grader og radian, **27**
 polære og rektangulære koordinater, **30-31**
 tid- og vinkelmål, **26-27**
 Omvendt Polsk Notation, **32**
 Omvendt registeraritmetik, **44**
ON
 decimaltegn, valg af, **60**
 nulstilling af den kontinuierlige hukommelse, **62**
 selvtest, **259-260**
 tænd/sluk, **18**
 Overløb, 45, **60**, 99

P

P/R, **64**, **66**
 Parenteser, **38**
 Pause PSE, **66**
 Permutationer Py,x, **47**
 Pi, **24**
 Polære koordinater
 kompleksstatus, **131-133**
 reelstatus, **30**

Potensopløftning y^x , 29

PRGM indikator, 64, 81

Procentfunktioner, 29

Procentisk forskel $\Delta\%$, 29

Program

afslutning, 66, 76

dataindlæsning, 67-68

etiketter, 65, 76

flytning af programpil, 81, 85

indirekte programkontrol, 106, 108-110

indtastning, 64-66

kørsel, 66-67

løkketællere, 108, 111-113, 115

programmeringsstatus, 64, 66, 85

registrering, 64

start, 67

stop, 66, 77

Programhukommelse

automatisk splitning, 214-215

generelt, 65, 68, 73, 214-216

sletning, 65

Programlinier (instruktioner)

definitioner, 65, 72

indsættelse, 82, 86

sletning, 82, 86

Programpil, 65-69, 81-82, 85, 89, 96-97, 104, 108-109

Programudførelse

efter $\boxed{\text{GSB}}$, 100

efter $\boxed{\text{GTO}}$, 96

efter overløb, 99

efter test, 91

fra eller ved hjælp af etiketter, 76-77

Pseudotilfældige tal $\boxed{\text{RAN\#}}$, 48

Påskeliljer, eksempel, 43

R

$\boxed{\text{R/S}}$, 66-67, 90

$\boxed{\text{RAD}}$, 26

Radioisotoper, eksempel, **92-93**

$\text{Re} \geq |\text{Im}|$, **122**, 125

Reciprok $\frac{1}{x}$

almindelig reciprokverdi, **25**

matrixregning, **148**

Reduktion, **230**, 231, 234

Registeraritmetik

direkte, **43**

omvendt, **44**

Registerkonvertering, **212-214**

Rektangulære koordinater

komplekse tal, **131-133**

reelle tal, **30**

Restmatrix, **157**

Resultatmatrix, 145, **146**, 148, 150

Returhop, **100**, 104

Rul-ned $\text{R}\downarrow$, **34**

Rul-op $\text{R}\uparrow$, **34**

Rullestak

aktivering af stakløft, **207-208**

bevægelser, **32**, 33-37

blokering af stakløft, **206-207**

fald, **32**, 37, 38

imaginærstak, **118-123**

indhold ved brug af $\text{R}\hat{A}$, 194, **199**

individuel behandling af matrixelementer, **144-145**

kompleksstatus, **129**

løft, **32**, 36, 38, 44, **206-208**

manipulationer, **33-34**

matricer, **172-173**

SOLVE , **179**

running, 67, 145, 180

Rækkenorm, 148, **174**

Rødder

eliminering, **230**, 231, 234

meningsløse, **186**, 188

R_0 og R_1 ved matrixregning, 141, **144**, 173

S

 $\boxed{\text{SCI}}$, 58

Sekant, eksempel, 101

Selvtest, 259-260

Serviceinformation, 260-264

SIDSTE X

besparelser med, 209

definitioner, 35

konstantlager, 39-40

korrektion af statistikdata, 51-52

matricer, 172-174

 $\boxed{\text{SIN}}$, $\boxed{\text{SIN}^{-1}}$, 26

Sinusintegral, 195-196

Skalaroperationer, 149-151

SKIP hvis SAND, 109

Skridtfunktion $\boxed{\text{SST}}$, 81, 84

Skæringspunkt på y-aksen, 54

Sletning

blinkende lyspanel, 99

forvalgtaster, 19

komplekse tal, 123-125

lyspanel, 21

overløb, 45, 60

sletteoperationer, 20-21

statistikregistre, 49

 $\boxed{\text{SOLVE}}$

algoritme, 180, 184-186, 217-219, 227-228

beregningstid, 235-236

diskontinuiteter, 224

generelle oplysninger, 178-179

ingen rødder, 184-186, 196, 227

konstante funktioner, 185, 186

lagerkrav, 190

minima, der er forskellige fra nul, 185

multiple rødder, 231

nødvendige forudsætninger, 217-219

nøjagtighed, 219-223

programmering, 189-190

rekursiv brug, **190**

restriktioner på brugen, **190**

specifikation af nøjagtighed, **235-236**

startværdier med 179, **186-189**, 218, 230, 234

stykkevist kontinuerte funktioner, **224**

ugyldige matematiske operationer, **185**

Splitning af lager, 42, **210-216**

$\boxed{\text{SST}}$, **81**, 84

Stak, se Rullestak

Stakken som konstantregistre, 39, **41**

Standardafvigelse $\boxed{s}$

population, **53**

stikprøve, **53**

Statistik

fakultet, **25**

gammafunktionen, **25**

kombinationer, **47**

korrelationskoefficient, **55**

lineær regression, **54**

lineært estimat, **55**

middelværdi, **52-53**

permutationer, **47**

sandsynlighed, **47**

standardafvigelse, **53**

statistikregistre, **49-50**

Statusindikatorer, **59**

Stjerner, eksempel, **40**

Summation af statistikdata $\boxed{\Sigma+}$, **49**

Systemflag, 91, **99**

T

T-register

$\boxed{T}$, **199**

generelt, 32, **33**

matricer, **171-173**

Talindtastning

afslutning, **22**, 36, **206**

- kompleksstatus, **119**, 123, 125, **126-127**
- TAN**, **TAN⁻¹**, **26**
- Tastkoder, **73**
- Temperaturspecifikationer, **264**
- TEST**, **90**
- Tilbagebetaling af lån, eksempel, **94**
- Tilbageføring, **85**, 89
- Tilfældige tal
 - generelt, **48**
 - lagring og fremkald, **48**
- Tilfældige tal **RAN#**, **48**
- Transponering, **148**, 149, 152
- Trigonometriske funktioner
 - komplekse argumenter, **119**, 132
 - reelle argumenter, **26**
- Tællere i programløkker, 97, **111-113**

U

u, **173**

- Udbedring af mangler, **260-264**
- Udlæsningsformat
 - definitioner, **57-58**, 60
 - indvirkning på **β**, **197**, 238, 240, **242-245**
- Underløb
 - generelt register, **60**
 - registeraritmetik, **45**
 - SOLVE**, **220**
- Underprogrammer
 - eksempel på kald i flere niveauer, **102**
 - grænser, **101**, 104
 - niveauer, **101**, 104
 - returhop, **100**, 104
 - SOLVE**, **178-179**, 189
- USER-status
 - generelt, 67, **77**
 - matricer, 141, **173-174**

V

Vektoraritmetik med statistikfunktioner, **56**

Vinkelmål, **26**

X

X ombytte med Y $\boxed{x \geq y}$, **34**

X-ombytning $\boxed{x \geq}$, **42**

X-register

$\boxed{f}$, **199**

definitioner, 32, 35, 37, 42, 59, **206-207**

imaginært, **207, 208**

matricer, 139, 154, **172-173**

$\boxed{\text{SOLVE}}$, 179, **181**, 189-190, 223

Y

Y-register

$\boxed{f}$, **199**

definitioner, 32, 37

matricer, 139, 154, **172-173**

$\boxed{\text{SOLVE}}$, 179, **181**, 189-190, 223

Z

Z-register

$\boxed{f}$, **199**

definition, **32**

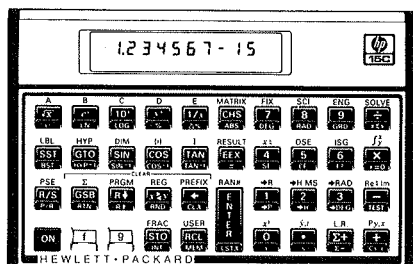
matricer, **171-173**

$\boxed{\text{SOLVE}}$, 179, **181**, 189-190, 223

Æ

Ændring af programlinier, **82**

tastatur og den kontinuerlige hukommelse



RULLESTAK

	Reel	Imaginær
T		
Z		
Y		
X		

Lyspanelet udlæser reelt X-register

SIDSTE X

----------------------	----------------------

DATAREGISTERBLOK

R_I

R ₀	0		R ₀	10
R ₁	1		R ₁	11
R ₂	2	n	R ₂	12
R ₃	3	Σx	R ₃	13
R ₄	4	Σx^2	R ₄	14
R ₅	5	Σy	R ₅	15
R ₆	6	Σy^2	R ₆	16
R ₇	7	Σxy	R ₇	17
R ₈	8		R ₈	18
R ₉	9		R ₉	19

Med startsplitningen er R₀ til R₁₉ udlagt som dataregistre.

Splitningen kan ændres med

.

Hvert register er på syv bytes. Der bruges én eller to bytes pr. instruktion. Når der bliver behov for ekstra programplads, konverteres dataregistre ét ad gangen til hver syv bytes programhukommelse idet dataregisteret med den største adresse konverteres først.

FÆLLESBLOK

R ₂₀	
R ₆₅	

Matrixhukommelse

Imaginærstak

og

Frle registre

Programhukommelse
Indtil syv
programlinier pr.
register

Splitningen udføres automatisk *inden for* fællesblokken.

Med startsplitningen er R₂₀ til R₆₅ udlagt som fællesregistre, som kan bruges til de avancerede funktioner og programhukommelse.



Hewlett-Packard A/S:
Datavej 52, 3460 Birkerød, tel. (02) 81 66 40

Hewlett-Packard Norge A/S:
Østerndalen 18, 1345 Østerås, tel. (02) 17 11 80